

Your

An Angus Specialties Production

OCTOBER 1984

80p

NEW

COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

**MACHINE CODE C8
TOO MUCH?
FIND RELIEF INSIDE...**

**LEARN ABOUT
BASIC WITH OUR
GREAT IN-DEPTH
SERIES**

**GAMES AND
UTILITIES
TO TYPE
IN**

**ADDICTED TO
ADVENTURE?
READ TALES FROM
THE CRYPT**

EA

9F

D2

16

**DOWN TO
BUSINESS:
CBM8296
HARDWARE
REVIEW**



**PAGES PACKED WITH
SOFTWARE REVIEWS**

DON'T JUST SIT THERE - PLAY SOMETHING!

FORBIDDEN FOREST

Forbidden Forest is made of a quest than just a game! The action takes place in a four dimensional scrolling forest landscape which many have entered, but none has emerged. He, I told my FOUR dimensional day fades into night as the action unfolds! The quest is to seek out and destroy the Demogorgon, mystic ruler of the Forbidden Forest. Before you can even set eyes on him you will have to contend with his army of fearsome creatures, including mutant spiders, swarms of giant frogs, snakes, dragons, skeleton soldiers and more! You have only your trusty bow and arrows to depend on!



£5018

AZTEC CHALLENGE

A challenge on an epic scale! Aztec Challenge takes you on a journey to Mexico and the ancient pyramid of Teotihuacan. The ancient Aztec gods and their devotees have ensured that no ordinary human can learn the secrets of the temple and live to tell the tale. The pyramid is protected by all

winner of treacherous traps and hidden perils - an epic test of your courage and cunning. Aztec Challenge features no less than seven totally different screens - here are just three of them - each of which presents a brand new challenge. We hope your joystick can stand up to it!

£5019

SLINKY

Slinky, the spring, was having fun hopping about when suddenly he came upon a pile of all coloured blocks, so he thought he'd play around on them for a while. Much to his amusement he found that they changed colour when he landed on them. Well, but unknown to him, the blocks belonged to the Wizard Wizard, who sent his friends along to take our poor hero. Slinky is a real fun package with ninety-nine levels, amazing reward displays, and action replays. Where else could you meet such charming characters as Dotty the dust devil, George the magician, Ralph the random roadrunner, and Lorenzo the chameleon hopper?



£5020

ON CASSETTE £8.95

ON DISK £12.95

FOR THE **commodore** 

Audiogenic LTD

P.O. BOX 88, READING, BERKS.

SEND FOR FREE COLOUR CATALOGUE!

Our COMMENT

**Our Editor has taken
line out from the
daily drudgery of
journalistic life (and
generally enjoying
himself) to introduce
his new magazine.**

CONGRATULATIONS! You have had the good sense and judgment to pick up a copy of the first issue of a GREAT new magazine dedicated to the Commodore range of microcomputers. If you have actually purchased this copy and are now sitting in your armchair at home you can rest assured in the knowledge that this and future issues of Your Commodore will satisfy your thirst for information, games, serious software, education, news and all sorts of goodies that are part and parcel of the Commodore scene. On the other hand, if you haven't already parted with money to buy this copy — Why not? We can assure you here and now that it will be money well spent — you'll gain an invaluable insight into your Commodore micro and what it can do for you!

What can we offer?

Assuming that you have spent a couple of minutes flicking through this magazine and you are still not convinced that Your Commodore is the best thing since sliced silicon, spend a little more time in my company and let me try to change your mind...



Your Commodore will entertain, inform and educate you on all matters 'a la Commodore'. Each issue will have regular news pages to keep you informed of all the latest products and stories related to your computer; our integrated software reviewers will be let loose each month on the latest packages around and let you in on their opinions before you actually hand over some cash; *Business* will be run around the spirit of *Adventure*; and at least one major piece of hardware will be reviewed each issue.

Commodore are not exactly renowned for the quality and clarity of their manuals — we have series and articles on programming in both BASIC and machine code to help you write programs yourself and to help you understand the way other programs are written — whether you are a beginner or expert there's sure to be something here for you to learn!

For the average home computer user (although we of course accept that Your Commodore readers

will be well above 'average'), games occupy a tremendous amount of the time and energy spent on the computer. Your Commodore will cater for the games player, as evidenced by our feature on games programming on the VIC and the fanzine games for the VIC and Commodore 64 in this issue. However we are trying not to go 'game-mad' and we appreciate that there are an awful lot (sorry, not meant literally!) of users out there in Commodore-Land who have exhausted their trigger fingers and cracked all their spacepal!

Seriously, though...

In Your Commodore we have put together a variety of articles for the more serious-minded amongst our readers; there are some really useful routines that are primarily intended for the Commodore 64, although they will run on other Commodore micros with a few alterations. The business users of Commodore micros have not been forgotten, either. Each issue of Your Commodore will have a number of pages devoted to the growing and generally under-served application area. — under the highly original title of 'Down to Business!' As you can see in this copy, we have a review of the CBM 8296-D and we take a look at how to set about writing your own business-type software.

Not only...

It would be true to say that we could go on ad infinitum about the value, expertise and entertainment that you are going to get from Your Commodore, but...

All you really need to know is that if you have a Commodore micro (regardless of which model) and you want to keep informed on the latest happenings in the world of Commodore, then just buy, read and inwardly digest each fantastic issue of Your Commodore.

Now pay less money and enter the fascinating world of Your Commodore!

Passing thought

Graham Davies, one of the merry contributors to this first action-packed issue thought that the following idea was worth passing on — why not link up your 50-64 to your normal video recorder? You can then use your TV screen instead of that tiny 3" screen. It works, you know!

But wait...

Should you consider that there is something you would like to see in a future issue of Your Commodore, why not drop a line to the Editor? We obviously put a lot of blood, sweat and tears into putting together a balanced magazine, but why don't you let us have your opinions? Write to the Editor at the Editorial address in London.



CCCCCCCCCCCC



VOLUME 1
NUMBER 1
OCTOBER 1984
Special Editor: Wendy J. Polak
Advertisement Manager: Mike
Seymour
Advertisement Editor: Gordon
Sue Cochrane
Managing Editor: Bob Harris
Chairman: Jim Connell
Originals: Penny Spinning
Design: MJD Design

Editorial & Advertisement Office
The T. Golden Square,
London W1R 3AB
Telephone: 01-437 5425
Telex: 887100

Your Commodore is a monthly
magazine appearing on the first
Friday of each month.

Distribution by: Argus Press
Sales & Distribution Ltd, 11 to
Paul Street, London EC2A 4PS
Printed by: Aldenham Press Ltd
& Son Ltd, York, Yorkshire
York

Subscription rates open
application to your
Commodore Subscriptions
Department, Inform Ltd, 1000
House, 175 The Mallway,
Hemel Hempstead, Herts, SG9
7SL

The contents of this publication
including all articles, designs,
plans, drawings and programs
and all trademarks and other
intellectual property rights
therein belong to Argus
Specialist Publications Limited.
All rights are reserved by the Law
of Copyright and other
intellectual property rights and
by virtue of international
copyright conventions.
Specifiably reserved to Argus
Specialist Publications Limited
and any reproduction without
the prior written consent of the
Publishers © 1984 Argus Specialist
Publications Limited.

MASTERING MACHINE CODE 6

The mysteries of machine code are explained in this tremendous series.

BATTLE ATAK 10

The drums are sounding and you are at war! Can you defeat the enemy before he gets you?

MACHINE CODE TO BASIC 13

A great little utility that does exactly what you would think — converts machine code to BASIC!

BASIC FACTS PART 1 14

BASIC is supposed to be such a simple language but few of us know and understand everything. We start at the beginning...



ARTISTIC STUFF 17

Fancy your chances as an artist? Our reviewer takes a look at a package that's almost as useful as a new paint brush.

CODE BREAKER 18

Do you consider yourself endowed with a logical mind? See if you can beat the computer at a great game of deduction and guesswork!

RENUMBERING 22

If your programs end up looking like a piece of numerical spaghetti, this useful routine could be just what you need.

SPRITE DESIGNER 24

Be speedy and do fantastic things with your Commodore 64.

FAST TAPE SEARCH 28

A great way of speeding up the Commodore tape system.

SOFTWARE SPOTLIGHT 30

Our fearless reviewers have been let loose on some of the latest software around for the Commodore 64 and VIC 20. So read on before you spend out some money...

YOUR OWN ARCADE GAME 40

So you think you can do better than the games you buy? We give you some general guidelines on writing your own!

CONCORDE II 43

Fasten your seatbelts and prepare to land the fastest aircraft in the world — Concorde II.

TALES FROM THE CRYPT 46

Don't worry — it's not as frightening as it may sound! Bureaucracy has been conjured up and has expanded on the spirit of Adventure.





INTRODUCING MODEMS 48

Modems have become rather fashionable in computing circles just lately... we let you into the secrets of their appeal.

MAKING MUSIC 50

You are chord-lilly (it) limited to make learning music that much easier with the use of your Commodore 64.

INPUT/OUTPUT 52

Some commonly asked technical enquiries answered in easy-to-understand language.

TRACK KING 54

Drive yourself crazy with this racing game for the VIC 20.

LADDER MAZE 56

Can you climb to great heights and get lost in the intricacies of this VIC 20 game?

REFERENCE LIBRARY 58

We've taken some of the Commodore books off our shelves and let you know whether they're worth reading.

VIC GAMES PROGRAMMING 61

We know there are thousands of VIC gamers out there and here we help you to write your own games.

COMMODORE 64 EXPOSED 64

We've taken more than a leaf out of this great book from Melbourne House and give you a taste of things to come from within the covers.

RELATIVE FILES 70

Relative files can be somewhat of a problem for some of you — but be confused no longer!

DATA STATEMENTS 73

All the news and information for Commodore owners.

DOWN TO BUSINESS 75

DOING IT YOURSELF 80

You shouldn't be put off writing your own business software — it's certainly isn't impossible and we try to show you how.

IS BIGGER BETTER? 82

The Cbm 804-D is an attractive looking machine. Does its performance match its appearance?

SUPERBASE OVERVIEW 86

Freeman's Superbase has received a lot of exposure — what does our 'expert' think of it?

CONTENTS



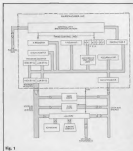
Although it was noted above that machine code is much faster than BASIC, it would be quite wrong to attach too much significance to it. If a particular program runs fast enough written in BASIC, there is little point in re-writing it in machine-code. On the other hand, there may be certain parts of a BASIC program which run far too slow for comfort.

For example, writing a large lot of data into order can take several minutes or, in some cases, even hours in BASIC. A machine code **polywriter** could achieve the same result in a few seconds. Screen animation effects are far faster in machine code than in BASIC. The movements are less jerky and the display can be given a more professional appearance in machine code. Moreover, the code is more compact, making it easier to fit on a computer's hard disk and machine code control of the robot's private parts is virtually a necessity — BASIC is too slow and clumsy.

Entering Machine Code

There are two methods — one is free but someone is using and terribly error prone, the other is pleasant to use and far less error prone but involves additional expenditure. The cheap method is by using the keyword **POKID** — see simply **POKID** every machine code instruction into memory, a byte at a time. Trying to follow that, a **POKID** machine code program doesn't, or to debug it is if it doesn't work. Nothing short of sheer hell! To compare the difference, consider back to the above example which added two numbers together. The labelled version, although its code-line is far more than shown later to be far more comprehensible than the current cross on the right.

By using a piece of software known as an **assembler**, the task is made easier, the machine coding is easy to read, correct and you are relieved of much boring work. Although this saves volume for readers who are



100

content to PCMC every-
thing, all programming
examples will be given in
assembly language. It can-
not be too strongly empha-
sized that serious machine
code work demands the
purchase of a good as-
sembler. We have used the
Mikrod assembler, which is
obtainable as a plug-in
card from:
SUPLESCH,
Winchester House,
Canning Road,
Wimbledon,
Harrow,
Middlesex HA3 7SL.

Hardware Knowledge Required

of "hardware knowledge" means familiarity with electronics, then you may not assuredly enjoy water handling. We shall, however, come to know, by maintaining "tools" but only in printed form. That is to say, if there a certain voltage is present or is not present. Computers happen to be electronic in nature, but, as far as programming in machine code is concerned, it wouldn't make all that much difference if they ran on North Sea gas. However, you will have to understand a little about the overall system of a computer, particularly the role of the microprocessor.

memory chips and a few other bits and pieces — not how they work but what they do. Figure 7 gives a rough idea of how some of the components are connected in most microcomputers. The microprocessor looks a bit bizarre at this stage but all should become clear as the series continues.

Many makers will already be aware of the distinction between RAM chips and ROM chips. ROM stands for Read Only Memory and refers to memory chips which hold permanent information put there by the manufacturers. Information in ROM is known as

ROMs are said to be non-volatile, meaning that the stored information is still there after the machine is switched off. There are three ROMs in the Commodore 64:

• The MC operating system ROM known as the **kernel**. This is a program which handles all the mundane operations of the computer such as reading the keyboard, displaying characters on the screen and so on. It occupies the range of hex addresses between 1000 and 1FFF (1024 to 8192 decimal).

- The 8K BASIC language interpreter occupying the hex addresses A000 to BFFF (16K to 49151 decimal).

■ The character generator ROM, responsible for arranging the correct pattern of screen dots for every character. There are two separate sets of 256-character: one set covers the upper case and fixed keyboard graphics, the other set covers the normal upper and lower case typewritten-style characters.

RAM stands for Random Access Memory, an unfortunate title, because ROMs chips are also random access, but constant. One of RAM's capabilities is the ability to read existing information and write new information under computer control. You may notice from Fig. 3 that the control line, *R/W*, is connected to the RAM chips but not the ROMs. This is because ROMs are permanently in the "read" state, whereas RAMs must be switched to a different state when information is to be read from them when there are to be written to. RAMs are more like the old school desks used in the Charles Dickens era — old material can be rubbed off and new material written.

RAMs are volatile. Any stored information is lost if the power is interrupted, even for a fraction of a second. RAM is for programs, although some of it is hogged by the operating system for screen display purposes. Input/output control and, for want of a better term, working space

There is indeed 64K of RAM present in the Commodore 64 but, unless some jiggling around is done (explained later in this series) and some ROM facilities sacrificed, only 128K is normally available to the user. It is difficult to proceed any further with computer components to machine code with the binary counting system as explained together with the base octal notation.

Unfortunately, the mere mention of binary is enough to make a sweat from those

who already know it and a gap of fear from those who don't.

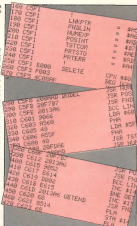
The Binary Number System

There are two kinds of computing machines, the analogue kind (which is of no interest whatsoever to us) and the digital kind. So, when we speak of 'computers', it is taken for granted we are referring to the digital kind. A computer is essentially a system of switches — silent electronic versions of the ordinary household on/off switch. There are hardly any smoothly varying voltages. Voltages are either in the HIGH state (about 5V) or the LOW state (nearly zero volts). Due to this essential two-state nature of the computer, it is natural to base all arithmetic and other forms of processing on a reasoning system which uses only two characters. Binary is such a system because it only uses the characters 1 and 0, allowing us to represent a HIGH voltage by 1 and a low voltage by 0.

Any number, however large, can be formed by a string of 1's and 0's called bits. Binary, like the familiar decimal system, uses the normal place-weighting system but, instead of each place being worth 10 times the value of the number on its immediate right, it is only worth twice as much. For example 111 in binary is read as one 1, one 2 and one 4 which, in decimal, is 7. To help you get the feel of binary, study the following binary numbers with their decimal equivalents:

1111 = 15
1010 = 10
1110 = 14
0011 = 3
1111 1111 = 255
1000 0000 = 128
1000 0011 = 131

Note that when there are more than four bits, it is conventional to separate them into groups of four because it is easier for humans to read them that way.



```

1111 0100 1100 0000 1010
 F  S  E  O  A
0011  0101
 3      5
  
```

Changing Hex To Decimal

Each hex digit is worth sixteen times as much as the digit on its right. In other words, the system is based on powers of 16 instead of the normal 10 as in decimal. The place weightings are shown below, in exponential as well as in decimal form:

```

164 163 162 161
(4096) (256) (64) (1)
  
```

Examples based on above weightings:

Hex 11F = $4096 + 256 + 16 = 4368$ decimal.

Hex 1032 = $4096 + 32 = 4128$ decimal.

Hex 21AF = $(2 \times 4096) + 256 + (16 \times 16) + 15 = 8832$ decimal.

Hex 0177 = $(1 \times 4096) + (7 \times 256) + (7 \times 16) = 5151$ decimal.

After glancing at the above "not-very-looking-jumble", it may be a relief to learn that only occasionally will it be necessary for you to convert hex to decimal. In fact, you will gradually realize that decimal is alien to machine code. It is actually easier and much quicker, to think in hex, particularly if you do most in an assembler.

For example, in machine code work, machine addresses are frequently involved. In hex, these addresses within the range 0000 to FFFF hex, in decimal, the same range is 0 to 65,535. Decimal is clumsy and quite unsuited to machine code: hex is concise and takes made for it. Memory locations are each one byte "wide" so the binary constants can be expressed very easily by two hex characters. The last address in memory is FFFF. There is something final about FFFF but there is nothing final about 65,535!

It does take a little time to get hex into your blood. In the initial stages it comes as a shock to discover that, "10" in hex is not ten — it is sixteen. Similarly, "32" in hex is not thirty-two — it is fifty. In speech, don't speak of 16

hex as "ten". Call it "one zero" to avoid confusion.

Adding Hex Numbers

Some examples are given below without explanation as an exercise:

13	3F
12	82
2B	51
5F	CD
01	1A
183	12

Binary Arithmetic

The pencil and paper procedure for adding binary is the same as for decimal, providing you realize that a carry to the next digit is worth 2 instead of 10. For example:

3	0011
add 5	0101
total	1000

Users of BASIC are usually unaware of the computer's internal arithmetic. Numbers of enormous size are casually entered and the correct answers are taken for granted. When we first take a look into machine code arithmetic, the situation looks decidedly bleak: the 6818 microprocessor is only capable of handling eight bits — in fact, it is called an "eight-bit" chip. If all eight bits are binary 1's, the largest absolute number it can handle is FF hex or 255 decimal. This may seem a depressing view.

If both positive and negative numbers are to be handled, the situation is even worse because one of the bits is used to indicate the sign of the number. This reduces the maximum positive number to only 127 decimal and the maximum negative number to -128. However, things are not so bad as they appear because, as we shall see later in the series, it is possible to increase the range of numbers by employing some crafty programming tricks.

Two's Complement Notation

Two's complement notation is employed in nearly all computers for the following reasons:

- It enables positive and negative numbers to be handled in an efficient manner.
- It simplifies the hardware concerned with addition and subtraction. Only a circuit capable of adding binary numbers is required; subtraction is achieved by adding the two's complement of the number.

The most significant bit in the byte, called the **sign bit**, has the following significance:

- The sign bit 0, if the number is positive, and 1 if it is negative.

Examples of positive numbers are given under the following table:

Some examples follow of Two's complement.	
Original	0111 1111
Two's complement	1000 0001
Original	1100 0000
Two's complement	0100 0000
Original	0000 0001
Two's complement	1111 1111

0000 1001 = 09 hex = 9 decimal
0111 1111 = 7F hex = 127 decimal.

Negative numbers are not so straightforward because it is not just a question of changing the sign bit. Before we give the rule, try and work it out by studying the following examples:

0000 0001 = +1
0000 0010 = +2
0100 0000 = +64
1111 1111 = -1
1111 0110 = -10
1000 0000 = -64

The term "flip" in the following rule means change "1" and "0"s and vice versa. To find the equivalent negative, flip all the bits and

then add 1 more. Ignore any final carry. For example:

0000 0111 = +3
Flip all the bits ... 1111 1100
Add 1 1
1111 1101 = -3

When we flip all the bits, the result is called the **one's complement** but, adding the final 1, converts it to the **two's complement**. The process works both ways: the rule still applies for changing a negative number back into positive. Try using the rule to change -3 back to +3.

To avoid adding the 1, which may often involve the propagation of a carry, there is an easier way. Assume we start with the positive number:

To obtain the two's complement, start from the right and copy down up to, and including, the first 1. Thereafter, flip the remaining bits.

The first rule we gave is the academic version, the second is the useful one — it is easy to use. Sometimes, we need to find the two's complement of a hex number. The easiest way is to write out the binary bits, use the second rule and put the result back into hex. For example:

If hex = 0001 1111,
Two's complement = 1000 0001 = C1 hex.

This concludes Part 1 of the series and most of the boring background knowledge. We could have stopped a lot of it but this would only have caused trouble later on in the series. A working knowledge of binary and hex is essential for machine code programming.

Get to be a real

Commodore and keep

the high seas safe with

this game from Jamie

Clyde.

BATTLE ATAK

BATTLE ATAK IS A computerized version of the popular game "Battleship." It runs on the standard Commodore 64 in no more than 10K of memory and uses user-defined graphics.

The scenario is as follows: you are in command of a submarine equipped with torpedoes and a battle computer. Your task is to defend your fleet of surface ships with your torpedoes and destroy the enemy fleet before their torpedoes wipe you out. You are shown your ships on one grid and where your shots hit on the other. Unfortunately you are unable to determine the locations of the enemy vessels, so the second grid starts off blank at the beginning of the game. Both fleets are identical and consist of the following:

- | | |
|--------------------|-------------|
| 1 Aircraft Carrier | (4 squares) |
| 2 Battleships | (3 squares) |
| 3 Submarines | (3 squares) |
| 4 Cruisers | (2 squares) |

The figure stated after the ship is the number of squares which it occupies on the grid.

The computer marks up on the grid any developments as the game progresses and the game will finish when either side's fleet has been completely destroyed.

When you run the program

You will first be shown a title page while the computer loads in the user-defined graphics into memory. To proceed, press the spacebar when told and you will be asked for a skill level and then speed level. Level 1 is the easiest game, and beyond level 4 the computer

will automatically have another shot if it has a ship. As far as speed levels are concerned, "1" is the fastest and "5" the slowest — this governs the length of time between each shot.

Once you have entered the desired levels, the grids will be displayed. But be-

fore the game can commence, you have to place the ships on your grid which is left of the coordinate in the computer, three characters must be entered. The first is the letter for the distance across, and the others for the figure for the distance

down — for example, the coordinate for the top left-hand corner would be "A1".

So, to enter a ship on the grid you type the coordinate where you want the bow of the ship to appear, and you then must say if you want the ship to be facing





A wondrous utility

from Mike Roberts

Converts machine

code into BASIC

for those of you with
problems.

MACHINE CODE TO BASIC

WHILE I WAS WRITING THE renumber program, often when in this issue it became necessary to make a BASIC listing from an area of memory as a list of DATA statements. Also I was doing some work with character generators at the time so I also needed a program that could read the character generator ROM as well (very difficult).

The result is a program that asks you for the start line number of the resulting program and the increment between the lines. Program lines will then begin to flash on the screen. After a while the program will end and the original program will be a lot longer. If you specify a start line number that the program already uses it will overwrite the program and will inescapably crash.

The start location for the code to be converted is stored in line 28 (currently 49153) the end location is

stored in line 55 (currently 49348-1).

The end result is the original program, which can

now be deleted and a long list of DATA statements containing the code. A POKE can be put as the first

line to POKE in the DATA (see line 18 of renumber) and key prints them you have it.



The conversion program

```

1 DEF FNHX=X/INT(X/256)
2 DEF FNL(X)=INT(X)-(INT(X/256)*256)
3 DEF FHX(X)=PEEK(X)+PEEK(X+256)
10 INPUT "CONVERT START NUMBER, INCREMENT":I,Z,X
20 POKE6304,PEEK(56304)+I*256:POKE1,PEEK(1)+I*256:REM START LOCATION
30 GOSUB1000:PRINT "P,Z,I" DATA "444:PRINT "00T000"
32 POKE6302,FNL(Z):POKE630,FHX(Z):POKE634,FNL(X):POKE635,FHX(X)
33 POKE636,FNL(X):POKE637,FHX(X)
35 POKE199,3:POKE631,19:POKE632,19:POKE630,19
40 POKE1,PEEK(1)+I*256:POKE5634,PEEK(5634)+I*256:STOP
50 POKE6304,PEEK(56304)+I*256:POKE1,PEEK(1)+I*256
51 DEF FNHX=X/INT(X/256)
52 DEF FNL(X)=INT(X)-(INT(X/256)*256)
53 DEF FHX(X)=PEEK(X)+PEEK(X+1)*256
55 I=PEEK(632):X=PEEK(634):C=PEEK(635):Z=Z+1:IF C=240 THEN GOTO REM END LOCATION
60 POKE1,PEEK(1)+I*256:POKE5634,PEEK(5634)+I*256:END
100 REM**
110 FOR I=6100 TO PEEK(X)+1: A=A+STR$(I)+", " NEXT A:LEFT$(A,LEN(A)-1)
120 C=C+10:RETURN

```

Like any language,
BASIC can be used
well or badly. A P
and D J Stephenson
brush up your
grammar.

THE BASIC FACTS PT. 1

BEGINNING A NEW SERIES on BASIC for the 64, or indeed any machine, is a problem. If the standard is pitched too high, new converts are disadvantaged; if too low, the growing mass of knowledgeable readers are inclined to get a little touchy and start muttering things about their intelligence being insulted. Another problem is the Commodore User Manual supplied with the CBM 64. If it was badly written, it would be easy to write an improved version. But it isn't. However, it is a rather thin volume and some users may feel that some of the subjects are given too sparse a treatment for comfort.

It is hoped that this series will help to fill some of the gaps. A certain amount of repetition is inevitable — that is to say, certain descriptions of the BASIC keywords will be little more than a re-statement of the Commodore User Manual. However, repetition is not always valueless: providing it is not literal repetition. Sometimes, the same facts expressed differently can change darkness into light — or at least twilight. However, it would be pointless to begin this series by defining or describing the purpose of every keyword in the BASIC vocabulary. It is expected that the reader has at least glanced through the Commodore User Manual and that it is kept available as a reference text when following this series.

What is BASIC?

If we define 'language' as a means of communicating, then BASIC is a primitive language. It is a set of rigidly defined keywords which, by



virtue of a special translation program, can communicate our orders to the computer. The list of keywords is the vocabulary of the language and the computer will recognise no others. It is impossible to overstate the fact that computers have no intuition whatsoever. If you use a word that is nearly right, the computer's meiotic intellect is immediately over-stretched so it gives up. The same thing happens if a colon is used when the thing experts a semicolon.

Anything you write must be precise with exact spelling and exact punctuation. The computer doesn't really understand anything — at least not in the normal meaning of the word. It is you, the writer of the program which expects intelligence. Computers themselves may start exhibiting primitive intelligence when the new Third Generation machines are launched but, for the moment, we must resign ourselves to the unpleasant fact that all computers, including the CBM 64, have zero intelligence. In spite of this, computers do seem to obey Stephen 3rd's fifth law which reads, "If a program has a flaw, a computer will delight in finding it".

The translation program is essential because the

computer, or rather the microprocessor chip within the computer, can only respond to a lower level language called 'machine code' (which is covered elsewhere in this issue). As far as BASIC is concerned, it hasn't got a clue. The translation program in the CBM 64 is buried within a ROM (Read Only Memory) chip. Each line in your BASIC program is translated (the technical term is **interpreted**) by the ROM into machine code before it is executed.

This takes place at electronic speed which, in most cases, is so fast that you may be completely unaware of the invisible translation going on behind the scenes. The translation is said to be 'transparent' to the user. And yet it does take time — particularly when your program contains mathematical operations which loops (parts of program which are executed over and over again until some exit condition is satisfied), loop times (milli-seconds) add up to large times (minutes or even hours) if there are enough of them so the translation time can, in some programs, reach an unacceptable level.

BASIC is not without its critics. In fact it has become fashionable to ignore quarters to deride the language. Some of the criticism is justified: BASIC can be slow at times and it is not easy to write programs which obey the rules of good structure (structure is a set of programming guidelines designed to make programs easier to read and debug). There is also an element of pedantry in some of the criticisms. BASIC was designed for ease of learning



and there is little doubt that the original aim has been met. It is the extent of all computer languages to owners of microcomputers owes a debt to the originators, John Kennedy and Thomas Kurtz of Dartmouth College, USA.

It has been said that the microprocessor brought computers to the Aigh street. This may be true but without BASIC, it is doubtful if the men or women in the high street would have bought many of them. The main languages COBOL, FORTRAN, ALGOL and Pascal have many virtues but simplicity isn't one of 'em. Of course, if you are not interested in programming and want the computer as a tool to run professional software or to play games, the computer language is of no consequence — but then this series will not interest you anyway.

BASIC Dialects

Native of London and Glasgow sometimes have communication problems. They both speak the same language but use different dialects. And so it is with BASIC. There are many variations on the original Dartmouth version. The essential features of the original language are preserved but there are some differences. Some dialects

may have additional keywords in the vocabulary or may employ different shades of syntax (grammar). This is understandable. BASIC was first launched in 1964 and the computing scene has changed almost beyond recognition since then. Hardware changes, such as high resolution graphics, joystick controls, graphics 'games', sound synthesizers and speech recognition chips, all need appropriate activating keywords.

The more sophisticated the translator, the bigger the ROM, the software in a big ROM costs more and, what is more important, uses up more of the **addressable space** of the computer. For example, the CBM 64 uses a 63/10A microprocessor. In common with most other eight-bit microprocessors, only 16 address lines are available. Limiting the number of directly addressable memory locations, including both RAM and ROM, to $2^{16} = 64K$. Apparently, CBM considered RAM space more important than ROM application so they decided to cut down the BASIC interpreter ROM to 8K. Many would say this is a pity. It is inevitable that penny pinching in this area has led to a rather stingy subset of BASIC, the best that can be said for it is that it is just adequate.

These goals (if structure should not be taken to mean. Complex programming projects can be undertaken whatever BASIC dialect is resident in ROM providing we work just a bit harder and for a little more methodical when we write programs.

Programming Guide Lines

If the CBM 64 is your first machine, and you have



ironed out the last remaining bug in your first program then, by all means, feel elated. But don't let the elation last too long. The fact that a program works should be considered the first, rather than the last, step. In all probability, the listing will be fouled by syntax errors resulting from multiple alterations and corrections. In all probability, there will be redundant lines and evidence of excessive zeal in the use of GOTOs.

It may be argued that, preparing a program works, why bother with structure? Your first program was probably short and structure won't have mattered very much. The trouble starts when your confidence increases and you have an urge to tackle more ambitious projects. If you carry on in the original undisciplined manner you will soon regret it. The bug-hunting stage becomes more and more tedious. The process of curing one bug often leads to the introduction of two more. The outcome, after hours or even weeks of work, can end in hours of verbal obscenities or, even worse, lead to permanent change in the personality.

The answer to all this is to start out on the right lines. Try and make the listing beautiful as well as functional. It is too early in the games to start laying down the precise rules of programming structure but, in the meantime, you would do well to take notice of the guide lines which follow.

Use Of Remarks

Use REMs liberally. It is often thought that remarks are for the benefit of other people trying to decipher how your program works. This is only partially true. They are more of an aid to the programmer. Never rely on memory for understanding the purpose of a routine. It may be perfectly clear at the time of writing, but even a few hours afterwards, you may not be quite so certain how it worked or, in some cases, even what it was

supposed to do! REMs use up memory but this doesn't matter in the slightest when you are developing a program. When it is finally finished, the original copy can be kept in its ROMed state and a copy, named of REMs, kept as a workhorse.

Preliminary diagrams

Sketch out a rough diagram, showing the order in which the various parts of the program are to be carried out. The type of diagram is unimportant to start with. There are things called 'flow charts' and 'top down structure charts' which use established symbols but there is no urgent need to learn, or even stick to them. Providing you understand your own diagram, that is all that matters. Too much establishment guidance applied too early can crush enthusiasm and destroy initiative.

Variable names

Decide carefully on variable names and write them down at a list. Unfortunately, CBM BASIC only allows two characters for a variable name, the first of which must be an upper case letter and the second a letter or one of the numbers 0 to 9. Some BASICs allow an unlimited number of characters in a variable name so it is easy to choose them meaningfully. For example, PLUS-ROBACE is meaningful but on the 64 we would have to be content with, say, PR instead.

Because of this restriction, it pays to spend some time on the choice of variable names. Although only two characters are allowed, the law of combinations allows you choose any one of 506 combinations. This can easily be demonstrated: the first character must be a letter, so there are 26 ways of choosing one. The second character can be a letter or a numeric figure so there are 36 ways of choosing it. The number of ways in which

two characters can be chosen is therefore $26 \times 36 = 936$ combinations.

Sometimes, it will be necessary to use a name without any meaning (no mnemonic value) in order to avoid clashing with another name. However, whatever names you give them, the most important thing is to write them down first and stick to them like glue. If you neglect this discipline, programming time will squandered by having to continually scroll



back the listing to see if you have used the name before. Giving the same name to (what should be) two different variables can cause distress to the programmer and pleasure to malicious computers.

Modular construction

As far as possible, try and arrange your program as a set of self-contained subroutines, each capable of separate testing outside the program. This is sometimes difficult but will pay dividends in the long term. If the subroutines require some values normally obtained from another part of the program (as is probably the case) then write a short trigger program which hands over test numbers, entered from the keyboard. When eventually, you persuade it to pass this test, it can safely be inserted in the program with the knowledge that, whatever else is wrong in the final version, that particular subroutine is beyond suspicion. In this way, your program grows in a coherent structure of tested building blocks. (The technical term is *modules*.)

The hated GOTO

Get into the GOTOs. The most violent critics of BASIC has always been directed against the overuse of the GOTO keyword. The GOTO is a jump to a line number: for example, GOTO 580 then, a few lines later, GOTO 1200. Too much of this is called 'spaghetti programming'. It can be absolute hell to follow if named to suit. GOTOs can never be avoided in old BASIC but, with a little care, their population can be diminished.

In any case, don't even think of jumping out of a loop with a GOTO before the normal exit. This is the worst crime in the book, even if the program does work. (There are one or two exceptions to this but this is not the proper time to discuss them.)

Screen messages

However brilliant the program is, effect on the user will depend ultimately on the quality of the screen display. There is only one thing worse than a scruffy display and that is one that is over-embellished. Messages to the operator should be as short as possible but not at the expense of clarity. Excesses should be avoided. For example, a message such as "ACF" is certainly short but it may not be immediately obvious that it is an invitation to input your age at the key board. Staccato messages of the kind often lead to a situation in which screen and user stare at each other blankly, each waiting for the other to do something.

On the other hand, "Would you be kind enough to supply the computer with information as to your age and then press RETURN" is an example of the other extreme and liable to induce generous vomiting all over the poor keyboard.

The CBI has a liberal assortment of graphic keys and there is a natural temptation in the early stages to lard up every message with a coloured border of stars, squares or asterisks. Over enthusiasm

in this respect can reduce the impact of the message by swamping it with the ill-developed beauty (?) of the border.

Programming philosophy

Always program with the idea that the finished product will, in the light of experience, require modifications or additions. This attitude will drive you to plan the program in module form. It will also provide the incentive to include plenty of REMs and, most important of all, it will teach you to cut down on the GOTOs and to keep a register of the variables used. In short, it will help you to write good programs — not just ones that work.

Finding And Curing Bugs

To start with bug finding before we have treated programming in detail may seem like putting the cart before the horse but it is based on sound psychological reasoning. Programming is unlike any other discipline because it demands a unique mixture of skills. Attempts to quantify the measure will inevitably invite argument, but for now let's write, programming is 10% science, 90% art. Imagination is also an important ingredient pro-



1. No program works first time
2. If it does, it was too simple anyway

There are two classes of bug.

• **Programming errors** which are acceptable to the computer. These are relatively easy to cure because the computer outputs a series message indicating the nature of the error. A full list of the error messages are given on pages 141-142 of the Commodore PET Manual.

• **Programmer errors** which the computer accepts but cause the program to behave other than the intended behaviour. These bugs are more serious and could take a long time to find.

The most common bug in the first group is due to the use of incorrect grammar. The computer obligingly informs the user that a "SYNTAX ERROR" has been encountered. The offence could be caused by an incorrectly spelt keyword, wrong punctuation, unrecognisable keyword, bracket (parenthesis) missing or one more than there should be, or the incorrect use of arithmetic operators. With regard to brackets, remember there must always be an even number of them. An incorrect arithmetic operator can be more difficult to locate, particularly if the question is at all complex.

The most common error is likely to arise over multiplication. When we write $Z = XY$ in ordinary mathematical notation, it is implied that X and Y are to be multiplied together. The computer does not respond to implications. It wants to know exactly, by using "=" as the multiplication operator, so we must write $Z = X * Y$.

It should be mentioned here that failure to include the asterisk in this example would not necessarily result

in a syntax error because $Z = XY$ is recognised as valid syntax. It will be interpreted as a single assignment, in which the variable named XY is to be assigned to the variable named Z. The example provides a good illustration of the second type of bug. That is to say, the computer accepts it as a valid order but, if the programmer intended multiplication, the result would be quite different to that intended. This kind of error can take some spotting until you get used to arithmetic operators.

While on this subject, it is worth discussing the concept of operator precedence. Page 134 of the Commodore User Manual gives the order of precedence in which arithmetical operators are carried out. For example:

$V1 = 3 * 4 * 5 / 2$ will make $V1 = 30$

$V1 = 3 * 4 * 5 / 2$ will make $V1 = 3$

When you are learning, it is safer to forget precedence and rely on parentheses to indicate the order in which the arithmetic is to be applied. The two previous examples would be safer written,

$V1 = 3 * (4 * 5 / 2)$

$V2 = 3 / (4 * 5 / 2)$

Although parentheses appear to be the inexperienced, they do increase calculation time so, in programs where execution time is critical, it is better to revert to precedence.

Syntax errors are responsible for much of the early frustration, mainly because humans are not used to dealing with such fussy animals. It takes some time to come to terms with such unaccommodating intolerance. Just BURN and RELOAD the program, peering away at each offending line, until the program eventually matches the last line number and then stops. The frustration should then be replaced by a warm glow of pride.



viding it is supported by a strong sense of method. But, above all, a programmer must always be on guard against the evils of frustration. It is ridiculously easy to be frustrated, particularly in the early stages of learning. The following advice (which should not be taken too literally) may prove convincing.



Our intrepid reviewer, Sue Duffield, was very impressed with this artistic program.

ARTISTIC STUFF

WHI COULD A COMPUTER teach me what nearly ten years of schooling couldn't, namely to paint? Could it and how? Meet our Master — how come Pampas.

Pampas is a complete painting and drawing system developed by the New Zealand company Kuma in conjunction with Kuma Computers.

Claimed by the makers to be capable of producing outstanding results, I actually worked through the 58-page manual provided to see if I could. Thankfully the manual is very definitely written in layman's terms — it even tells you how to load the contents which is a boon for the simple-minded. It gently takes the user step by step through the basics of the program, always working by example and you quickly reach the point where you actually start to paint.

The touch (and seven required) and the question and answer type entry bring you straight into the drawing mode.

And there you are — Pampas's seascapes, Cornish-style landscapes or even Picasso's madness — all at your fingertips.

The detail

So how does it work?

When the program is loaded, a crosshair appears on the TV screen. You begin to paint by moving the crosshair across the screen using a joystick or a series of singular keys. A trail of minute dots follow wherever direction you choose to go: left, right, up, down, and even diagonals. Three painting methods are available: pen, brush or text mode.

Using the pen mode there are around 32,000 dots in the painting area



from which you make up your pictures. This means your painting can be both intricate and delicate but at the same time it is both painstaking and slow — not recommended for the slipshod among you.

The brush mode means you can colour large areas of the screen quickly and these are individually definable brushes and brushes which you can store and remove at will.

The text mode is, as the name suggests, simply the ability to reproduce on the screen any of the keys, letters, figures and symbols, accessed through the keyboard.

You paint in four colours at any given time, chosen from the full colour range of the Commodore 64.

Once you start to paint, if you make a mistake or change your mind, you don't have to wipe the

screen clean and start again. Instead it wipes a manner of retracing your steps using a different colour. Who knows how Van Gogh's paintings would have turned out if he'd had this facility?

You can choose to do the hard work out of moving the crosshair round and round the screen as you paint because there are some 'home' positions, accessible by a single keystroke, which take away the necessity of going over and over the same area to get to where you want to be.

A little practice and in no time at all you'll be producing results to stun your friends. And you don't need to lose the masterpiece when you switch the TV off and go to bed. You can save all or part of your painting on tape and retrieve it when you want. The manual also gives

detailed instructions on how to photograph your efforts direct from the TV so you can keep them forever.

Very young children can play happily at this game, using the joystick, with no mind or washing up and few tears.

Adults can get equally hooked creating and recreating at little cost to either pocket or temper. Even professional artists claim the manual can use the system to produce accurate publishable visual work.

Truly versatile, totally addictive and just fascinating, this is one of the best 'games' I've played for a long time.

Paintpic is for the Commodore 64, is priced at £19.95 and comes from Kuma Computers.





Can you outguess the machine? Derek Waldron's programmed a C64 version of this popular game.

CODE BREAKER

THE PROGRAM IS A version of the popular type of code-breaking game. It's a game for two players who take the roles of 'code-writer' and 'code-breaker'. In our version for the Commodore, the computer is given the role of code-writer, leaving you the task of tracking the code.

The code consists of four colours from a choice of eight. Not only do you have to guess which colours have been used, but you also have to determine where each colour has been placed within the code. To help you reach the goal, the computer will mark each guess that you make, according to the following rules: for each colour and position guessed correctly, a black 'peg' will be awarded. Where a colour is guessed correctly, but the position does not match that in the code, a white 'peg' will be awarded. A mark of 4 black 'pegs' therefore means you have cracked the code!

In view of this assistance, you are only allowed 10 attempts in which to break the code. If, after this time, you have been unsuccessful in your attempts, the computer will reveal to you what the code was.

To enter your guess, keys 1-8 should be used, the colour chosen is pressed, a square on the grid will be filled with the appropriate colour, working left to right as each colour is entered. It should be noted that once a colour has been entered it cannot be deleted, so careful thought should be given before entering any colour.

Once the fourth colour has been entered, the computer will automatically



draw your guess and print the appropriate mark in the answer box alongside the line you have entered. The answer will be in the form of black and/or white 'pegs' in accordance with the rules already specified. Please note that the position of any black or white 'peg' in the answer bears no relation to the position of the colour in your guess. This means that a black 'peg' in the left-most position of the answer does not necessarily mean that the left-most colour of your guess is the one that is right.

Watch Out

The relevant points to watch for in each answer given by the computer are the number of 'pegs' and their colours.

If any of your guesses are totally unsuccessful, obviously no pegs will be awarded in the marking, which will be backed up by an unpleasant noise. Do not be discouraged. You will come to learn that an unsuccessful guess will often give you more information than one rewarded with, say, one or two 'pegs'.

The code for using sound within a program is limited, and has been restricted to the already mentioned 'code noise' for an unsuccessful guess, together with different pitched notes for black and white 'pegs', when marked. When you are able to crack the code within the 10 guesses allowed, you will also hear a short, tinkling tune.

A summary of the instructions to play the game are included in the program.



The Program

Line 14	Clears the screen and changes the background to grey, with a different shade grey border	Line 803	Checks if the answer is all black 'pigs'
Lines 20-90	Prints the first page defining the object of the game	Line 810	Checks if the answer is nearly wrong
Lines 100-110	Requests a key to be pressed to continue	Line 820	Sets up a loop to be repeated but as many times as there are 'pigs' to be printed
Lines 110-140	Prints the second page giving instructions to play	Line 830	POKEs a circle (arg) directly to the screen
Lines 150-160	Requests a key to be pressed to continue	Line 840	Checks if any black 'pigs' are to be awarded
Line 170	Sets up the initial values for the variable array C(n)	Line 850	Decrements the black 'pig' counter by one, POKEs the 'pig' position with the colour black, goes to the 'black pig note' subroutine
Line 171	Defines A()	Line 860	GOTO 880
Line 172	Defines the variables used for producing sound	Line 870	POKEs the position of the pig with the colour white and then goes to the 'white pig note' subroutine
Line 180	Sets up the initial values for the variable array A(n)	Line 880	Repeats the loop
Line 190	Sets C() equal to a	Line 890	Prints a black pig on the answer pad and goes to the 'play a tone' subroutine
Lines 200-430	Clears the screen and prints the grids on which the guesses and answers are placed	Line 900	Asks if you want another go at the bottom of the screen
Line 500	Generates a random code of four colours stored in the array C (n)	Line 910	Asks for a key to be pressed
Line 540	Increments C() by one and checks if 10 colours have been made	Line 940	Checks if the key was a 'Y' and replaces the game if it was
Line 510	Clears C() and sets up a loop which will be repeated 4 times	Line 950	If the key pressed was not an 'N' the program goes back to request a key
Line 540	Requests a key to be pressed	Line 960	Ends the program
Line 545	Converts the key pressed on an ASCII code, which is assigned to the variable C()	Line 970	Clears the 'answer go' question from the bottom of the screen and resets the program to Line 170
Line 570	Checks that the key pressed is allowable keys 1-4	Line 1000	Sets volume to maximum, sets waveform to pulse, sets high and low pulse rates
Line 580	Converts the key pressed to a number representing the colour chosen, which is then assigned to the variable C(n)	Line 1010	Sets high and low frequency values, sets up sustain value
Line 590	Repeats the above loop until C(n) contains 4 values representing the colours of your guess	Line 1020	Play note
Line 600	Clears the flags used in checking your guess	Line 1030	Resets high and low frequencies and sustain value
Line 610	Sets up a loop to be repeated four times	Line 1040	Plays note and then turns voice off
Line 620	If guess does not equal code, repeat loop	Line 1100	Sets volume to maximum, selects waveform waveform, sets attack value
Line 630	Sets the relevant guess and code flags and increments the black 'pig' counter by one	Line 1110	Sets high and low frequency, and sustain level
Line 640	Repeats the loop	Line 1120	Plays note and turns voice off
Line 650	Sets up two loops used in checking out of position colours	Line 1200	Sets volume to mid level, selects waveform waveform, sets attack value
Line 660	Checks if the flags have been set	Line 1210	Sets high and low frequency, and sustain level
Line 670	If guess does not equal code, repeats the loop	Line 1220	Plays note and turns voice off
Line 680	Resets the flags and increments the white pig counter	Line 1300	Delay loop
Line 690	Repeats the loop	Line 1310	Sets volume to maximum, selects waveform waveform, sets attack level
Line 700	Goes to the 'print answer' subroutine and then goes back to the request input for next guess	Line 1320	Repeats note values and duration from data statements
Line 750	Assigns variable C(n) which is used as a POKE address for printing your guess	Line 1330	Checks if all notes have been read
Line 760	POKEs directly to the screen a 'reversed space'	Line 1340	Sets high and low frequency of note
Line 770	POKEs the relevant colour to the position POKEd with a reverse space in the above line	Line 1350	Sets sustain value
Line 780	Repeats from subroutine	Line 1360	Plays note, turns voice off, goes back to play next note
Line 800	Assigns the addresses to which the answers are to be POKEd	Line 1370	Data containing values of notes to be played in tune
		Line 1380	Updates the position in the array A(n)-A(n)
		Line 1390	Prints the code if you were a successful cracking it
		Line 1400	Gets the colours of the code
		Line 1410	Delay loop before returning to ask if you would like another go.

should be able to invent commands recognised by your own machine quite easily.

Deleting colours

Should anyone feel that it would be desirable to have the ability to delete a colour before all four have been entered and marked, the following may be of use to you.

Amend line 545 to read: 545 GC=A5C(C5)-IF GC=38

THEN 1608

Invert these lines: 1608 IF 3=1 THEN 150

1608 388-1: POK% GP(X)=429,12:POKE GP(X)-440, 1

1610 GP(X)=GP(X)-88: GOTO 545

This will only work on the line you are currently entering, providing the fourth colour has not been entered.

Hint

The program should be easy to convert to other home macros. Reversed characters within the quotes will be one of the following:

- Reversed H: Clears the screen and places the cursor in the home position (top left of the screen)
- Reversed V: Places the cursor in the home position without clearing the screen
- Reversed R: Turns the reverse video on
- Reversed O: Turns down one position
- Reversed |: Cursor right one position

```

400 GOTO
410 IF (C1) THEN GOTO 410:IF (C2) THEN
420 GOTO 410
430 GOTO 410
440 GOTO 410
450 GOTO 410
460 GOTO 410
470 GOTO 410
480 GOTO 410
490 GOTO 410
500 GOTO 410
510 GOTO 410
520 GOTO 410
530 GOTO 410
540 GOTO 410
550 GOTO 410
560 GOTO 410
570 GOTO 410
580 GOTO 410
590 GOTO 410
600 GOTO 410
610 GOTO 410
620 GOTO 410
630 GOTO 410
640 GOTO 410
650 GOTO 410
660 GOTO 410
670 GOTO 410
680 GOTO 410
690 GOTO 410
700 GOTO 410
710 GOTO 410
720 GOTO 410
730 GOTO 410
740 GOTO 410
750 GOTO 410
760 GOTO 410
770 GOTO 410
780 GOTO 410
790 GOTO 410
800 GOTO 410
810 GOTO 410
820 GOTO 410
830 GOTO 410
840 GOTO 410
850 GOTO 410
860 GOTO 410
870 GOTO 410
880 GOTO 410
890 GOTO 410
900 GOTO 410
910 GOTO 410
920 GOTO 410
930 GOTO 410
940 GOTO 410
950 GOTO 410
960 GOTO 410
970 GOTO 410
980 GOTO 410
990 GOTO 410

```

CODE

BREAKER

Mike Roberts has
come to the rescue of
tangled line numbers
with this great
renumbering utility
for the C64.

COMBODDOR BASIC is lacking in a great deal of functions. Graphics commands have been well documented and most magazines have published programs that do hi-res, sprites, sound etc.

Unlucky tend to be a bit late on the ground as everybody's efforts are directed against the graphics. While using SuperSoft's 'Video' assembler I suddenly needed a renumber program. The following machine code program was the result. As I only use a lot renumbering assembler programs I'm afraid that there is no facility for renumbering GOTO/GOSUB etc as this would make the program quite long and complex.

As it stands it is just as good as the renumber program in Commodore's Simon's BASIC cartridge.

Setting Great Store

To explain how the program works it is first necessary to go into detail about how BASIC stores a program line. The first byte in a program line is zero (not the last byte as most people seem to think), the next two bytes contain the address of the start of the next program line. This is called the link address and is used by the interpreter only while running a program and inserting a new line. The next pair of bytes is the pair that we're interested in — the line number. After the line number is the rest of the BASIC line.

The line number is stored as a 16 bit binary unsigned integer, so it can be from 0 to 65535. So for a renumber program all we have to do is step through the program poking out the

RENUMBERING RENUMBERING RENUMBERING RENUMBERING

1234567890E
1234567890E

Machine Code Program

```
10 B=0C000
20 START=0FF
30 INCR=0F0
40 PTR=0F0
50 COUNT=0FD
60 LDX START+1
70 STX COUNT
80 LDX START+1
90 STX COUNT+1
100 LDX #000
110 STX PTR+1
120 LDX #003
130 STX PTR
140 LDY #000
150 NEXTLN LDX COUNT
160 STA (PTR),Y
170 JSR INPT
180 LDX COUNT+1
190 STA (PTR),Y
200 CLC
210 LDX COUNT
220 RDC INCR
230 STA COUNT
240 LDX COUNT+1
250 RDC INCR+1
260 STA COUNT+1
270 LOOP JSR INPT
280 LDA (PTR),Y
290 CMP #000
300 BNE LOOP
310 INY
320 LDA (PTR),Y
330 CMP #000
340 BNE HKT
350 INY
360 LDA (PTR),Y
370 CMP #000
380 BEQ ENDT
390 HKT LDY #000
400 JSR INPT
410 JSR INPT
420 JSR INPT
430 JMP NEXTLN
440 ENDT RTS
450 INPT INC PTR
460 INE NEXTLN
470 INC PTR+1
480 NEXTLN RTS
```

zero bytes, indicating the start of a line, skip a few and modify the two bytes containing the line number — easy!

Of course there are some options. You can decide the line number that the program starts with and the difference between them. This is done by POKEing four bytes into the 545 memory:

```
POKE 545, INT(0/250)
POKE 547, 0-INT(0/250)*250
```

This will set the start number for the program where '0' should be replaced by the number that you want the program to start with.

```
POKE 550, INT(Y/250)
POKE 552, Y-INT(Y/250)*250
```

This will set the difference between the lines, where 'Y' should be replaced by the increment.

The machine code should then be called with 555 45112, after a short pause the program will be renumbered. The POKE10 values will be preserved, so you only have to POKE10 them the first time, the next time they will still be there and you just need to do the 555 call.

BASIC loader

1 REM RENUMBER BY MIKE ROBERTS JULY 1994

```
10 FOR I=49130 TO 49240: REDEF: POKE I, 0: NEXT I
20 DATA 146, 247, 134, 289, 168, 248, 134, 254, 162, 0
30 DATA 134, 252, 162, 3, 134, 251, 168, 0, 145, 253
40 DATA 145, 251, 32, 00, 192, 165, 254, 145, 251, 24
50 DATA 165, 253, 181, 249, 133, 253, 165, 254, 181, 250
60 DATA 133, 254, 32, 00, 192, 177, 251, 281, 0, 268
70 DATA 247, 208, 177, 251, 281, 0, 288, 7, 208, 177
80 DATA 251, 281, 0, 248, 14, 168, 0, 32, 00, 192
90 DATA 32, 00, 192, 32, 00, 192, 76, 18, 192, 96
100 DATA 238, 251, 208, 2, 238, 252, 96, 255, 255, 255
```


EPainting in the pixels
was never easier!
Design your own
sprites with this
package from
J McHale
and A Carter.

SPRITE DESIGNER '64



TYPE IN THE PROGRAM provided and use it on a blank cassette. Now RUN it and if all has gone as planned, a message will appear on the screen, telling you to place a blank cassette in your C24 unit so that Sprite Designer '64 may be SAYED to tape as a working code file.

After it has been SAYED, verify it to ensure that there are no errors present.

To load 'Sprite Designer '64', type 'LOAD', 1, 1 — then type 'SYS (64738)' to reset the Commodore 64's pointers before using the program.

As 'Sprite Designer' is located at \$C000 (4096 decimal), you may use it in conjunction with a BASIC program of up to 128.

Sprites designed by this program are located between pages 230 and 254 inclusive ie (240×64) — (253×64) + 63.

You may design a total of 25 sprites which should be sufficient.

SYS (3000) initiates the sprite colour.

Movement is by means of the cursor keys, which I found to be the most

practical, with the Shift key playing its usual role ie (2478 = Right; 2558 = Left).

When initialised, 'Sprite Designer '64' displays a grid and four sprites on the right hand side of the screen, with the markings 'X', 'Y', 'Y' & 'x/y' displayed alongside.

These letters stand for standard sprite, X — expanded, Y — expanded and X&Y expanded respectively.

Current sprite status & Page No. is displayed at the top of the screen and you should see a cursor flashing in the top left hand corner of the 24 × 21 grid.

Every bit set in the sprite block will be displayed on the grid as a 'W' and every blank bit, a 'V'.

Remember to be careful when using the 'Clear grid' function as it also wipes the sprite definition.

It is a good idea to familiarise yourself with the keys before starting to define sprites.

BLUR/SLOW — Rotors has been disabled for programming purposes but is re-enabled on exiting to BASIC.

Useful Locations

\$C0A7 — \$1967 is the only location to use for the user. It is used in the timing of keyboard response etc.

The value of this location is set at 40.

Changing this value will make key response faster or slower, 1 = fastest; 255 = slowest.

Explanation:

Function	Description	Key Used
Clear Grid.	Clears sprite grid and current sprite block.	CLR
Page Flip.	Advances to next sprite page.	↑
Page Minus.	Loops to previous sprite page.	↓
Exit.	Exit back to BASIC.	ESC
Save.	Save sprite definitions to tape.	SAV
Enable Multicolour.	Enable sprite Multicolour mode.	MC
Disable Multicolour.	Disable sprite Multicolour mode.	DMC
Change Sprite colour.	Change colour of sprite.	CHC
Change colour Reg 1.	Change colour of register 1.	CH1
Change colour Reg 2.	Change colour of register 2.	CH2
Rotate Sprite.	Rotates sprite through 180° (horizontally).	ROT
Invert Sprite.	Rotates sprite through 180° (vertically).	INV
Fill.	Fills in one bit of the sprite definition.	FILL
Delete.	Deletes one bit of the sprite definition.	DEL

[illegible][illegible]



This 64 utility from
Alike Roberts will
really speed up your
tape system.

FAST TAPE SEARCH



First on

My program is designed to be recorded as the first program on a tape. The procedure when wanting to load a program is to rewind the tape to the beginning, load and run the directory program last, and this is important, DO NOT PRESS THE STORY DIRECT BUTTON. When the program runs it will ask you what file you want, and whether you are going to save it or load from it. Next comes the important bit — you must hold down the PLAY key and press down the F/WD key at the same time so that they are both down at once, now you must hold down the F/WD key and press the STORY DIRECT key. The end result of all this maneuvering is a silent tape recorder and a depressed F/WD key.

Now if you merely press the Return button, the tape recorder will magically start to make a whizzing sound. After a number of seconds, depending on which file you want, the tape recorder will stop and you can proceed as normal.

The system will find programs on tape a lot faster than any normal method.

But wait...

There are one or two restrictions though:

The tape must be blank when you start to save programs on it, you can't just tack the catalogue on the beginning of a tape of programs and hope it will find them. The program

expects the programs at evenly spaced intervals, that's how it works. The program is currently set up for 10 files of about 6K each on a C60 cassette. These parameters can be easily changed to suit your own needs.

The number of programs on the tape is stored in a

DATA statement in line 238. This must be equalled by the following number of DATA statements. The DATA statements are padded out with spaces so that if you add a new program name to the list and re-record it at the beginning of the tape it will be the same length as it originally was and not

overwrite the first file.

The length of the sectors is usually 6K and the length is dictated by the "10" in line 180. Double it for 16K sectors, halve it for 4K etc.

This is an invaluable utility and I have been using it on the PET, VIC, and now the 64 for about the past five years.



COMMODORE

```

30 PRINT "C"
40 PRINT "C"
50 PRINT "C"
60 PRINT "C"
70 PRINT "C"
80 PRINT "C"
90 PRINT "C"
100 PRINT "C"
110 PRINT "C"
120 PRINT "C"
130 PRINT "C"
140 PRINT "C"
150 PRINT "C"
160 PRINT "C"
170 PRINT "C"
180 PRINT "C"
190 PRINT "C"
200 PRINT "C"
210 PRINT "C"
220 PRINT "C"
230 PRINT "C"
240 PRINT "C"
250 PRINT "C"
260 PRINT "C"
270 PRINT "C"
280 PRINT "C"
290 PRINT "C"
300 PRINT "C"
310 PRINT "C"
320 PRINT "C"
330 PRINT "C"
340 PRINT "C"
350 PRINT "C"
360 PRINT "C"
370 PRINT "C"
380 PRINT "C"
390 PRINT "C"
400 PRINT "C"
410 PRINT "C"
420 PRINT "C"
430 PRINT "C"
440 PRINT "C"
450 PRINT "C"
460 PRINT "C"
470 PRINT "C"
480 PRINT "C"
490 PRINT "C"
500 PRINT "C"
510 PRINT "C"
520 PRINT "C"
530 PRINT "C"
540 PRINT "C"
550 PRINT "C"
560 PRINT "C"
570 PRINT "C"
580 PRINT "C"
590 PRINT "C"
600 PRINT "C"
610 PRINT "C"
620 PRINT "C"
630 PRINT "C"
640 PRINT "C"
650 PRINT "C"
660 PRINT "C"
670 PRINT "C"
680 PRINT "C"
690 PRINT "C"
700 PRINT "C"
710 PRINT "C"
720 PRINT "C"
730 PRINT "C"
740 PRINT "C"
750 PRINT "C"
760 PRINT "C"
770 PRINT "C"
780 PRINT "C"
790 PRINT "C"
800 PRINT "C"
810 PRINT "C"
820 PRINT "C"
830 PRINT "C"
840 PRINT "C"
850 PRINT "C"
860 PRINT "C"
870 PRINT "C"
880 PRINT "C"
890 PRINT "C"
900 PRINT "C"
910 PRINT "C"
920 PRINT "C"
930 PRINT "C"
940 PRINT "C"
950 PRINT "C"
960 PRINT "C"
970 PRINT "C"
980 PRINT "C"
990 PRINT "C"

```





We've searched the software shelves and thrown the spotlight on some available packages. See what our intrepid reviewers think of them before you part with your money.

SOFTWARE SPOTLIGHT



SHADES OF BATTLE ZONE this one! Confined with a 3-D plot (yes, it's 3-D), you hear the almost approaching Diamond shaped, but missing alien which don't see at first, appear and are easy prey. Then, if you have the sound up on your TV, you will hear at times a low humming noise. This is when your shooting accuracy counts. The noise signifies that there is a suitable alien ready to come towards you. One shot is enough but it's getting that one shot that is the problem.

Once you dispatch the alien a door or vortex opens on the screen. When you travel through it you will have to move round a star field making sure you don't hit anything. When you accomplish this minor (I) task another door opens into the second level. This level is a little harder than the first with the alien shooting back and alien which explode on their own.

Definitely a good game having eight landscapes and three skill levels, it is very challenging. Having the sound effect is quite helpful as they tell you what to expect from simple aliens to a blipping exploding alien on level four.

S.J.P.P.

OUR REVIEWERS HAVE spent long hours going over the software to be assessed. They have tried to bear in mind a number of factors that will be of interest to you, the reader. Such as the use of sound and graphics, how quickly the game became boring or addictive and to give you an overall impression of what

to expect should you decide to spend some of your hard-earned cash.

Each review has been given a star rating out of a maximum of five stars (meaning: brilliant, why haven't you rushed to your local store already) down to one star (which means that the alien and you won't be ecstatic with your

new purchase!). Always read the review and do not judge the package solely on the star rating — you may not be too bothered about the poor graphics and sound that have earned the package this opinion. And always remember that the review is only one man's opinion — one man's meat, etc.



YOU HAVE ENTERED THE mountain ridge in your helicopter determined to stop the trucks carrying the arms from getting through to the enemies' base. They have heat seeking missiles which they fire at you. This is the setting from Sunlock's game, Chopper. This is done by moving your chopper left, right, up and down, dropping bombs on the trucks.

The graphics are jerky, the sound is not very good

and the animation is fair. Loading is quite simple. It is an addictive game and very quick. The action is fast and it gets faster as the game progresses. It's a pity there are no skill levels. There are other good points especially the feeling you have when you have beaten the guided missiles. The scoring is very simple, 50 for a large and small truck, 100 for a tank and 250 for a missile launcher. The key choice is good and joystick response is also good. This is a brilliant idea for a game, however it's a great pity that it was not thought out more constructively. Overall a fun and reasonably enjoyable game from Sunlock but for £5.95 it doesn't appear to have as much value for money. P.W.W.



Masters

ORBITRONICS

£5.95

IBM - joystick (Optional)

THE IDEA OF THE GAME IS to go into a mine and collect the treasure in the smallest possible amount of moves. Presuming you begin doing this, it's perhaps the meanest bunch of mines a video gamer could hope to run up against, the Mines.

The game starts with your character clutching a sword ready to do battle with the Masters for the treasure, which is scattered somewhere in the mine. To enable you to fight, you have to be strong and to keep your strength up you will find there's food around the mine. You will also find prisoners in the mine who are willing to divulge the whereabouts of the gold. The other way to

find the treasure is by pressing 'V' which will display a portion of the real mine.

When you fight with the Masters you will find that you lose your sword if you kill it. This isn't a problem as long as there is a spare sword lying about, which there always is. If there are 5 then you may have to fight with your bare hands and then you stand a chance of losing your life.

It is worth paying attention to the graphics in the fight because of the intricate detail that has been put in. It is a good game with no two mines the same.

SLIP



Merchandise

PARADISE SOFTWARE

£5.95

IBM - joystick (Optional)

HAVING BEEN VARIOUS software companies make attempts at the game "Inramble" and bring home, not succeeding, I was quite surprised and excited at seeing this version.

Simply, it is even better than the original arcade machine game (yes, it's possible). To start with, the quality of graphics used in the game surpasses all other versions. The game sequences are kept as close to the original as possible with few exceptions.

The first is that when playing you will notice some

musical notes scattered on the floor of the landscape. When these and the music is turned on or off. This is a good idea, since the only drawback is having "Star Wars" playing in the background. This spoils the sound effects.

The second deviation from the original is screen 3. This screen contains mobile tanks that shoot at an angle, and it's quite difficult to line at them and maintain a good supply of lives.

It is a hard, if not harder, than the original but the only high score feature is when you're playing. It's definitely well worth the money as it is an example of State-of-the-Art games. Look forward to more projects from Nigel Rowland at Rabbit Software.

SLIP

Trials

UNIVERSITY

£5.95

IBM - joystick (Optional)

TRIALS COMES IN A BOX containing an end of instructions, feature listings and pictures of the game. I couldn't wait to grab my joystick and start playing. On the box it said there was "3D total perspective graphics". "Smooth multi-colour graphics, music and sound effects" etc, etc.

The game itself took a long time to load but I could wait. The description of the game was that the galactic

merchants had constructed mine mines in which they could travel. Unfortunately, the Truck kept on attacking without warning.

This is where your fleet of prizes come in. It is your job, firstly, to clear a patrol of Trucks who advance on you at an alarming rate (on the second level) and, secondly, to clear a path through the minefields. Once both screens are accomplished, you see a merchant vessel disappear behind a planet and land safely, then you return back to the first screen again!

The graphics on this game are not exactly the best, although the laser light is quite good as it is shown in perspective so you move.

SLIP

Alpha Blaster

JANUARY

£5.95

IBM - joystick (Optional)

THIS NEW GAME FROM Januak puts you in control of the latest space fighters, it

is a version of the arcade game Astroblaster. This version has no speech or warp factor. The basic idea is to shoot down everything in sight. This includes spaceships of different kinds and sizes. They do come down in a uniform array of movement and you have to dodge the

unbelievable asteroids. After going through the asteroids you clock into the motherhips; this is not a hard task because the computer automatically docks you. This is the basic idea of the game.

The sound is very good and the spaceships are in different colours. It is a fast

demanding game. A novice at shoot-'em-up games is in for a hard time and even an expert won't find it easy.

The key chains, 2 light C and 5 light to fire, is good and makes it easy to play.

My only criticism is a bug. Overall this is a very good, fast moving game.

PWW

SOFTWARE FRONT AT SPOTLIGHT



Scramble
★★★★
Atari
\$2.95
VIC 20 • (optional Expansion)

THIS IS A GAME IN THEIR "Frontier" Series. Once loaded, which is a bit long, the screen displayed the logo PLE and a bar movement up, left, right and down shift to fire and CTRL to bombs. Key C starts the game. The screen of the arcade game of the same name, graphics are clear and colour is reasonable. The game is written in machine code but is fairly slow, movements tend to be jerky, sound is used effectively and continuously and becomes

irritating. The idea of the game is to fly past spaceship by randomly through a series of caves, avoiding mountains, rocks, monsters and other hazards, using your lasers and bombs to destroy the enemy's missiles and fuel dumps. This is explained in detail on the colourful insert.

Considering this is for the unexpanded VIC 20 they have made good use of the memory. However, the end result is a game which is disappointing and soon becomes tedious.

It requires a difficult game to play and the lack of a high score or ball of time doesn't encourage you to keep trying to improve your skills.

PWB

Metabolite
★★★★
F32
CBM 64
\$2.95

Few Atari rippers seem to have the lasting appeal of good old-fashioned Space Invaders. However, as a variation on the theme, Metabolite will have you instantly hooked and as a piece of arcade action should have you practising in your sleep.

You are the sole defender of the metropolis which is under attack from waves after of kamikaze alien intent on destroying the city below you. Your space ship is in perpetual motion and you have to prove your prowess by dodging be-

tween all the alien craft and activating them with your superior fire power before they destroy all the buildings.

Apparently there are twenty four waves of aliens to defeat and naturally points for every alien craft you zap plus an additional variable bonus at the end of each wave depending on how many you hit. At wave 15 you enter the advanced stage and that's when they begin in on you as well as the city, pass onto the championship challenge at wave 19 and the termination attack at wave 22.

In short, it's fast, it's vigorous and it's energetic and I lost count of the variety of alien craft and the number of times it's broken my joystick. When you're in

Horace Goes Skiing
★★★★
Atari Home
CBM 64
\$2.95

HORACE IS NO STRANGER when it comes to computer games. Hungry Horace was his last skiing title. This time he's determined to display his prowess on the slope. Regardless of the quality of the game, one of the prime advantages is the use of a fast loading system which loads in less than two minutes. There's nothing worse than hanging around

for ages waiting for a game to load. But I digress. Despite a partial similarity in the initial stages to those frustrating 'Trigger' games, Horace Goes Skiing is an enjoyable and useful game. The objective of the game is to get that cheeky little character across an extremely busy main road avoiding the juggernauts, cars and motorbikes and into the shed on the other side to pick up his ski. To reach the ski slopes of the Hannekeas run you have to get Horace back across the road which gets busier as time passes. What happens

if Horace gets splatted? Well it's not the end of the game. At the start of the game Horace has \$40 in his pocket and every time he gets run over costs him 10 for the ambulance ride. It also costs \$10 for the ski to cross the road has a few chances. Extra cash can be accumulated by continually crossing the road for which points are gained and a \$10 bonus given for every 1000 points. Once on the ski slopes the fun begins as Horace has to ski between the red and blue flags, paralleling left and right and losing or gaining points

according to his scores. But there are obstacles. Strategically placed cones get in the way and if Horace hits one, there is a chance he will have to start all over again if he hits the cone. There are also clearly placed mounds to force Horace off balance.

At the end of the run Horace simply has to cross the road again to reach the next slope. Of course there is more to this game than immediately meets the eye and despite its apparent simplicity has a high level of lasting appeal.

E.M.





Flight Path 737

 Arcade
 27.95
 VIC 20 • MS-DOS • (joystick)

THIS IS ONE OF THE ONLY flight simulations I have seen for the VIC and proved to be excellent. You have a choice of 5 levels, including a free solo flight (easier) to test pilot (hardest). The

game has an insight clock, fuel indicator, altitude meter and many other dials found on the flight deck of an aircraft. On taking off you must have the flaps down and be at 180mph. The screen shows the runway which scrolls towards you in 3D perspective. The visual effects are good and are quite realistic. When airborne you see a

backdrop of mountains. The graphics tend not to be as good as the runway but are effective. Whilst cruising you may have to extinguish a fire in the aircraft. This is done by pressing key I. You now come to the difficult bit, landing. It starts with ease but the element of difficulty soon appears when an adjustment of your instruments is necessary.

Once landed, you are told how well you have, or have not, done and you are given a score accordingly. The sound is very realistic and adds flavour to the simulation. The game loads very quickly and is one in a series of flight simulations to come for the VIC. I look forward to purchasing the next one out.

PWW



Star Commando

 Terminal Software
 17.95
 CPM 64

STAR COMMANDO IS one of those games that looks like it will be an instant winner, but when I actually started playing there was something about it that was not quite right. I couldn't put my finger on it at first. Then it hit me: slap bang between the eyes. The thing you'd expect to be able to control, the gunnights, was rigid and unworkable and, instead, the joystick controlled the background. That was it. The game was ruined.

Nevertheless, onward in the face of adversity. As the rookie in the star fleet your mission is to repel the cosmic invasion force consisting of the Calibre Pirates and Sessonic Raiders.

and clear the space lanes of mines. Every sector of the galaxy has to be searched and cleared. The only problem is that your energy level is decreasing all the time and you need to find your mother ship to re-energise.

You do have an advantage, however, a long range scanner which can be convoked in any clear sector. The only problem is that the information is never complete. Points are scored for each sector cleared and each invader blasted. The more points you score the higher your ranking at the end of the game — Captain, Commodore or Star Commando only skill dictates but the last does give you access to the secret code enabling you to claim your wings from Terminal.

Overall impressions, well over concept, shame about the gunnights. R.M.



Castle Cavelon

 Ocean Software
 24.95
 CPM 64

WHO SAID THE AGE OF chivalry is dead? Who wouldn't come to the rescue of a damsel in distress answering her pitiful cries for help? So you fancy your chances do you? Then enter the castle of Cavelon. The only problem is that it's not the easiest of

tasks climbing your way to the top of the turret. Castle Cavelon is a stronghold guarded by an array of knights and archers who wander the stone-walled corridors eager to deter any intruders. To reach the top of the castle turret you have to ascend six levels of stairs each of which is hidden behind a doorway at the end of a maze of corridors. To open the door you have to have all the pieces of the door in your possession. Search the

corridor well because that is where they are hidden. Also floating around the corridors are a number of rewards. Collect them all if you can. Their usage will render you immune for a while from the knights and archers. There's no doubt you'll need them all if you are to claim your prize.

On screen data indicates your points total and the comparative achievement of your rival if you chose the two player option, as well as the percent of the door you

have collected, the number of lives you have left and how many magical talismans rewards you have managed to find. You'll not be surprised to discover that this is yet another variation on the illustrious Pernan theme. However, that is not to denigrate what is, in fact, a hold in its own right and is quite an enjoyable game. And anyway, who in their right mind would leave Cavelon crying in vain for assistance. R.M.



THANK GOD FOR THAT. Isn't real. If it was, either the airline would have run out of undamaged passengers by now or my exploits would have been turned into one of those sky-high Hollywood disaster movies.

Needless to say this is not

one of the recent games around but it has to be one of the more addictive. There's no giving up on it until what at first seems like the impossible has been achieved, even if it is only getting the jet safely off the runway and into the air without smashing the flap to smithereens or breaking the undercarriage off.

The trouble is, that is only the beginning. Even in the easiest level there's the mountain to get over and then there's the problem of getting the plane down

again safely on the runway without trying to tunnel beneath it. Like any flight simulator, Flight Path 730 requires a high degree of eye, hand, memory and general brain co-ordination.

As a representation of an aircraft flight deck, the screen is crammed full of flight information, heading, altitude, altitude, fuel, flap and undercarriage status and distance from home—including a view from the cockpit window.

Flying the 737 requires a combination of joystick and keyboard control plus a propensity not to panic. In the case, by some freak chance, you manage to Software Spotlight 8 cope with your first solo flight, there are five other skill levels to tackle, all of which progressively shorten the runway and increase the height of the mountains, whilst throwing in the odd on-board fire and last minute emergencies on

landing. Getting your wings won't be easy, so whatever you do, don't panic.

If I had to describe the game at all, and I feel really reluctant to do so, it would be on the basis that there is a tendency to master each level a stage at a time. If it's the landing you're always having trouble with, getting to that stage can be a bit of an unnecessary obstacle. The addition of a practice level would be a welcome solution.

RM



TO BE HONEST, I DIDN'T know what to expect. Once loaded, using Euro-byte's fast loader system called 'Overdrive', I was confronted by a screen bereft with a 'New Showing' sign including a piece of music the same of which is on the top of my longest VHS tape.

Now the game... Dan Lucas who is the Author of the game, even went to the extent of composing a poem in which there are clues to guide Hugo (that's you) through a labyrinth, frustrating graphics, arcade-type adventure. I spent the first half hour, with beads of sweat on my forehead,

trying to get past the first level!

The object of Level 1 is to get up the down escalator whilst trying to avoid the dummies coming down from the top of the screen. You have to go through the door and that is not easy. I had to read the second verse for the screen (no more clues).

Second level is a little more difficult, with yet more down escalators to go up! This time you have got three exits to go through. I went through the wrong one and had to start again... on Level 1!

It's a must for any computer game enthusiast who thinks he or she can play games well. It is well worth the money and you will be writing to Euro-byte for more clues when you can't figure it out. 3 P P



THIS IS A 16K TEXT adventure. You have crash landed on the planet Mars and the only way to get home is to find a missing part of your space ship.

The game is written in BASIC and with ingenuity you can break in and cheat from through written in BASIC, imperious are fast and this is not a problem. The instructions are brief and typed into the insert. There is no sound except for the introductory tune. The text is black on white and the cues, when on black the program gives a description of the location including obvious exit and

objects near by. Commands are given by usual verb noun combination although you can use one letter entry for N, S, E, W, U, and D. Pressing R gives a description of the situation that you are in. Pressing I gives you an inventory of the things you have, the game loads easily with a very snappy save on the side page. The game itself is difficult but that is partly due to limited vocabulary and BASIC language analysis. The problems themselves range from easy to difficult. There is no save to tape option like on many VN adventures.

There is a sense of humour. I found this adventure addictive. It caused me sleepless nights and many cups of black coffee.

P W W



A NEW GAME FROM Atari, and one of the best I have seen. You are Bongo the super mouse, always wanted to be a mouse. You have to rescue the princess and win her heart. If are definitely changing into a mouse. The only way to win the princess is to find the King's lost diamonds. Bongo set out and eventually arrives at the chosen location on the overbank. This is where you take over. You control Bongo with the joystick which is very intuitive to the sighter touch. The game uses the whole of the screen, the graphics are very large and superbly designed. The

game itself is in its part: like layout is King/Mario Minor type; in other words, walkways, ladders and trampolines and an added bonus of a slide feature which only you can go down (obviously the rabbits suffer from vertigo). There is a two-player option and three levels. There are six screens which makes it addictive trying to get to the next screen. The sound is very good with a catch tune at the beginning.

It does take a while to load the game but it is well worth waiting for.

All in all an excellent game and should remain a firm family favourite.

P. W.



THE PYRAMID, like MIDGE new series of software, comes complete with a high-speed loader to enable short waiting times for the player to play the game. Another nice aspect of this game is the abundance of instructions and information included. A high score form can also be found with the literature.

The program itself is about "Zigg" (that's you), making his way through 120 chambers of weird and wonderful structures, all of which are dangerous to touch. The way to get through the Chambers is by collecting crystals when they appear, and dropping

them on the forcefields which are guarding the next chamber. Zigg can only pick up the crystals when they are in the harmless state. If Zigg picks up a crystal before it is harmless then some of the shield power is used up. The shield power is always decreasing because of the alien ramping into you.

If you reach zero energy before leaving the chamber, the game ends with you exploding and, assuming you score more than 38,000 points, your score is given a code which can then be sent into the software company. If your score is high enough, they will put you on their 1000 top scores list published every 6 months.

Generally, it is a very good game with reasonable sound effects.

S.L.P.



ANOTHER SHOOT'EM-UP from Jodeltek. A superb game which has several screens and a very fast. Your task is to penetrate the enemy defences and destroy all aliens at each stage. The first screen takes you onto the rear area; these you can shoot. The next screen is the warp stage, these are very clever as they wait for your bullets to go past until they move. After this are the war-hoppers, these move diagonally down the screen and bounce off any obstruction except your bullets and you. The fourth screen are the missiles which come down and shoot you but then this is fairly easy as you can send up showers of bullets.



GOT A BIT OF A DODGY boiler have you? Fancy doing a bit of plumbing? Not really! Then why not get old George to do it for you. He's willing, one of the best ship fixes in the business and, what's more, runs around at your beck and call.

With the water temperature in the boiler rising rapidly and threatening to blow a stop cock at any moment, you have to get George to plumb a relief pipe from the valve at the bottom left hand corner of the screen to the boiler in the top right.

Easy? Well not quite. With time gently ticking away, you have to identify all the right bits of pipe at

The penultimate screen is by far the most difficult. Space battles arise in three shells until, for a brief moment, they come out and you get the opportunity to shoot at them. At last you reach the last screen. The Corner phobos's stand up and shower bullets down. The graphics on this screen are superb as the birds flap their wings. You are left with one final task which is to travel through the space corridor. This is easy and if you have managed to get this far you are granted a bonus ship. Back to screen one you are taken with an army of more furious aliens.

This really is a superb game and credit must be given to the writer's ability to fit all this into the unexpanded VIC.

P. W.



the bottom of the screen to weave the relief pipe around the boulders indiscriminately littered in your path. Mind you, if you do get stuck there's always that handy piece of dynamite to blast your way through.

But be on your guard. The ghost of George's former employer is on the rampage and dying to get him a bad name in the business by preventing him from completing the relief pipe in time.

Naturally the game gets progressively more difficult to the point where it is practically impossible to see the boiler for the boulders, so pick up your blow torch and get plumbing. R.M.

SOFTWARE SPOTLIGHT



Jumping Jack

Dorland
US \$5
VHS 26 • joystick/keyboard

THIS IS A VERSION OF THE arcade favorite Frogger. If you have not heard of it, the basic aim is to get a frog across a road, wading out for traffic. Once over the road you have to find your way to the big pond, to get there you must hop onto turtles and logs making sure that you don't jump into the water. The obstacles gradually move faster making it quite fun to play. Also in this version the graphics are very pretty and the ammonia leaves a lot to be desired. The choice of keys are good - forward/back, 2 left and right, this makes it very easy to play.

The joystick movements are also very sensitive.

The game is written in full 6800 machine code. While the game is being played there is a 'tone' which soon becomes irritating; fortunately you can always turn it down.

Unfortunately there is a 'bug'. It is possible to go off the bottom of the screen and reappear at the top without having to cross over the road or jump onto the logs or turtles - therefore the game is much easier to complete and high levels can be achieved with great ease.

This is a disappointing version of the arcade game but is one of the few versions of Frogger around for the VIC 20. It certainly won't wear out your trigger finger, but it does provide good family entertainment.

P W W

Zodius

AMIRCO SOFTWARE
US \$5
KEM 64 • joystick

ANYONE WHO HAS SEEN the game "Shamus" on the Atari will notice more than a passing resemblance with this offering from Amiroc. As with most other games, there is a desperate force on the loose. This time they come in the shape of Masters of Black Magic.

The Masters have stolen the twelve signs of the Zodiac and thrown them down in a 400 corridor maze, and it is your job to retrieve them and place the signs in the central room. That it is not as straight forward as that, because you have to fight the demon slaves which inhabit all the corridors. Be careful if you

re-enter a corridor because there will be another set of demons to cope with.

If the demons catch you or you run into a wall, your character is immediately vaporised and all that is left is your last.

When playing this game I did come across some interesting features. The first was that after I had reached a sign and had run into the next room, I found exactly the same sign again with some monsters! The second was a little bit more disappointing. As my character ran from one corridor to another through a door, I materialised in a wall and got dematerialised yet again!

It is a good game with very good space graphics but it doesn't have that certain little something that makes it truly addictive.

S L P

Chuckie Egg

A & P Software
US \$5
KEM 64

WHAT A PLEASANT surprise! For Chuckie Egg and Donkey Kong, I thought I'd suppose I'd need to go further! Perhaps I'd better.

You are Mr. House Harry and the object of the game is to guide him around the Mr. House

collecting the dozen eggs that have been laid. Yes, it's always a dozen and some are ingeniously suspended in mid-air. There is also a lot of corn to collect and all the while you have to keep old Harry out of the way of the ducklings which are busy gobbling up the corn and chasing Harry intent on pecking him to death. Willy ducklings and not chickens I don't know!

Anyway, just like Donkey Kong there are platforms for Harry to walk

along, ladders to climb up and down, and lots to bash a ride on providing you can get him to jump accurately.

There is a choice of six game speeds ranging from 'for those who prefer adventures' and 'for the faint hearted' through to 'for the general idiot' and 'for the suicidal maniac'. In addition, there are a variety of skill levels to progress through on each speed level but beware, on skill level one the mother duck is let loose to run riot.

There's nothing at all pretentious about this game. It's a good solid arcade action which follows a tried and tested formula which has proved very successful over the past year or so. But you need to know it for what it is otherwise, with the absence of any real description on the packaging, I suspect many a games addict would be disappointed in having bought something they already own but is burdened with a different title. R A





Gridmap

A +
Live Wire
\$8.99
C64M 64

FIRST IMPRESSIONS ARE rarely wrong and in the case of Gridmap it was pretty poor. However, before I get too negative I have to admit that the graphic representations are extremely good indeed. But, having said that, it is the execution of

the game that really counts and that's where Gridmap falls down miserably.

Despite some lengthy instructions, the game at first seems like a hopelessly overwinded mess with a tedious musical accompaniment. It takes a while to fully realize what the hell is going on.

The object of the game is to maneuver a character called Mr. Live Wire from box to box in the goal to reach and diffuse indi-

vidually placed bombs. Bombs are stored depending on how quickly you reach the bombs and how many bombs scoring (lagged) squares you take in on the way. There are dangers of course. The skull and crossbones are out of bounds and several big boom skulls around warning to kick you into touch. Once a square has been used it disappears from the screen although there is a facility to scroll the

remaining squares in a line to help movement around the grid.

After every series of five bombs is defused, a new screen is presented and the game continues at a much greater difficulty level. Obviously the idea is to amass the highest score. At least the two player option brings an element of human challenge into the game which suffers dreadfully from going to be too clever by half. **B**

Bathtime

A +
M4
\$7.99
C64M 64 • Jantick (1 or 2)

THESE ARE GOOD SILLY games and bad silly games. The trouble is that I can't decide which this one is. It is very subjective as to whether you like violent or non-violent games. Bathtime is a non-violent game, which has reasonably good graphics and sound.

The main feature is its large use of sprite graphics including an elephant, a boy in swimming trunks, two heavenly cherubs, a goldfish and a swim. The game itself has an interesting concept, as follows: The fish and swimmers have a bath and controlling the flow of water are the two cherubs; they, being you and another player, or the computer. It appears that

the two cherubs have had a little argument and are dead set on making life difficult for each other.

Player one in the two player game runs the water into the bath while player two runs the water out. Player one has to either let the bath flood so that the swim sink away or let the bath run dry and kill the fish. Player two has to prevent this by monitoring the water flow. There is a time limit on each game played so as to let player two win if both swim and fish are still present.

To make things a little more difficult there is an elephant which keeps on appearing warning a drink, and a boy representing down a bucket of water. The towers and raises the water level respectively.

A good game for the very young but not for the space insider enthusiast.

S L P P

Cybotron

A +
Amazing Software
\$7.95
C64M 64 (1 or 2 joystick)

SOME TIME IN THE FUTURE robots will become endowed with greater self-determination and will decide that humans are an unnecessary evil exterminate the weak blooded ones... Fortunately Cybotron — a somewhat up-rated human — can save the human race... or can he (you)?

There are several games with a similar theme on the market and although I can only get to level 7 with difficulty (there are 1000 there is the feeling of a reasonably balanced game — that with a little more luck (graciously), I could get further. Only joysticks may be used — no keyboard —

but there is the option to use two... one for movement and one for firing at the 'baddies'. Who says computing is a solitary pursuit? If you have time to examine the graphics you will find there are 64 different robots, 5 types of electrified stationary objects that you can destroy but must not touch, the solid having to rescue and of course your 'alterego', Cybotron. His scores are recorded and you have 3 lives to gamble with. Bonus waves, where there are a higher ratio of humans (no robots) to robots will help boost your score — and to test your shatterd nerves, there is a pause option. Cybotron has 'Turbofire' to cut down on loading time... and this worked perfectly every time. A game for the hard and 'shoot' brigade with quick reflexes — well balanced and worth a try... **P**

SOFTWARE FRENCH KIDS SPOTLIGHT

Dr. Watson's computer learning series — BASIC adventure part 1
Glenview Publishing Ltd
(ISBN 0025 76 2)
1985

THE "SERIES" ON THIS product has several — let us be generous — ambiguous statements. Let us look at these and try and decide what we have in our hands.

First: "The revolutionary new concept in computer learning provides a FUN way for children to experience the 21st century world of BASIC programming..." So far, so good — the book does present a novel teaching concept. Programs and exercises are introduced in a storybook manner that might well catch a child's imagination.

Second: "Part One of the book is an evening adventure" (our heroes must learn to operate the shop's computer if they are

to escape!" The new concept is an ambiguous but the word adventure to most computer owners implies that the user has to solve an adversary — not so, the book works its way carefully through examples of PRINT, INPUT and string handling, and in the last two pages of the story our intrepid Dr Watson, jury right a spaceship's controls — with no reference to our previous examples of computer programming — and sets it in motion. Not quite what we expected!

Third: "This unique adventure is specially designed to teach the fundamentals of 64 BASIC by way of example — the way children learn best." Well, perhaps so — but for the use of that word "adventure" again.

Fourth: "Accompanying this is a tape containing the same programs as are on the specially 1 computer, and the same 64 BASIC teaching

programs to your children can learn as they play." QUOTE Yes there are three sample programs that are referred to in the book — very short but yet in the book! We then have two copies of a "Hangman" program — a little odd but I maintain that one can have a lot of fun with these sort of guessing games — still with a total vocabulary of four (4) words! AND one of those 64 BASIC — not exactly inspiring, it also makes one wonder about who is kidding whom at the price of 29.95.

Having turned the tape over and I loaded the first program, we are presented with a menu. This lists the programs on the side of the tape and on selecting option 'a' you are given a very brief résumé of what you will learn... then check a "Message from Dr Watson" — this turns out to be pages of errors from the book! With my package there was

no indication that this was there — not even a cheap bit of paper...

The teaching programs request the user to type in the correct format for PRINTING various examples — followed by similar exercises on string handling and use of Line Numbers — quite a nice idea and well thought out... except... with a teaching program such as this one does not expect to be able to "check" the program if given a wrong answer — OK, I tried to do it but...

Fifth: "Every BASIC command covered is also given a separate careful explanation in the second part of the book..." Yes it is... except for the errors mentioned above, it's possible.

In conclusion, all I can say is... was for Mr Holmes's book — he always knew better than Dr Watson!

P.P.

Dragon-Box
Cassiopeia
© 1985
(Class 44 (copyright only))

THIS MUST BE ONE OF THE most frustrating games I have ever played for a long time! You are a lion (I'm trying to get out of a cavern by jumping from ledge to ledge, to reach the exit above you. The "power" of your jump is controlled by how long you hold the

joystick left (or right) before releasing the stick — then you jump. It takes a little while to get the hang of judging how much power to put into a jump. In order not to overshoot or undershoot your target, just to add to the frustration — there is a fierce dragon roving around the cavern, ready and willing to gobble you up! You have only one life and having lost that (I did, with monotonous regularity), you have to walk through the 18 second

"trap" as Dragon-Box falls into the cavern once again.

Time elapsed, Level and Hit-Points are indicated at the bottom of the screen. Level is the height you are at, above the floor of the cavern. Hi-Points does not indicate until you have escaped the cavern, so no matter how long you manage to evade the dragon, or how high you climb, you have no personal best score to strive for in the early stages... I could have done with a little something

to spur me on after being gobbled up for the fifth time! Graphics are superb, I just wish it was a bit easier at the start, to help inspire me to greater heights! There is no game option — but then if you don't last long enough to achieve a Hi- Score, you can always drink your coffee in the 18 second wait following death!

P.J.

Typing Wizard

Incentive Software
CMM 84

WHAT A SHARPING TITLE page — I always did like watching wizards throwing lightning bolts. ... out of the field of fire, naturally! What a pay,

one of the menu options is not "wizards recall". This program sets out to teach you touchtyping — a very commendable project, especially as I don't! I would guess it was written using a monitor, as the choice of colours for some instructions was strange letters on a medium-grey background — not the easiest to read, even if you turn the colour off at

the TV. Instructions show like this are few and anyway you want to get on with your typing so perhaps seven may be forgiven — a thought for the future could well be user-definable colours!

Menu driven, you are offered nine options — 1, the introduction from then on you are told how you should place your fingers

and thumbs for different rows on the keyboard — starting with the "home" row A-Z-X.

This sort of program is really only as good as your "will to learn" — if you have that "will", Typing Wizard can only help.

S L P

Space Pilot

ARCADIC SOFTWARE
CMM 84 • joystick (Optional)

PACKAGED IN THE NOW standard library case, Arnic's "Space Pilot" comes with a picture showing aircraft of different wars fighting each other.

In this game you are an ordinary 1984 Jet Pilot. The very fact that you are from that particular time is a definite help on the first 3 screens. In the first screen

you are faced with a 16 strong squadron of bi-plane fighters (the year being 1918, say you think, and now it is, for these pilots do not stand a chance against your gleaming 1984 fighter). All that is left to do on that screen is to tackle a single Zeppelin which comes floating across your screen, but be careful because it can release a nasty little salvo of bullets when you are not expecting it. Once you have cleared the screen, you are transported to another time — 1940. Again, you must dispatch a large number of aircraft. But they are getting a bit more nasty.

Clear that screen and

you are launched into 1970 where you meet helicopter fighters. This is the turning point of the game. As you clear that screen you are transported into 1980! The aircraft are more advanced than yours, so you have to play more tactically instead of just firing into them and shooting.

Finally, if you clear that screen you meet from the year 2001, Flying Saucers! Unfortunately I didn't clear that screen but I did naturally!

This game in general has good graphics using sprites, the sound has a few effects but not too much to write home about. With the time-



limit and other obstacles like the clouds it can lead to very interesting suicide runs.

S L P

Splat

Incentive Software
CMM 84
16/89

THIS IS A GAME FOR ALL potential members of the Strawberry Jam Preservation Society. This object of Splat, amongst other things, is to prevent the hero, Zippy, from being thoroughly pulverised or, more colloquially speaking, turned into strawberry jam.

To all intents and purposes, Splat is a variation on the age-old theme of the maze game. Computers can be drawn with Pacman, the ubiquitous voracious of all maze games, although it can

be a bit misleading. However, if the truth be known, it is just as addictive.

The main difference between Splat and other maze games is that the main character, Zippy who you control, is not being chased by an monster, at least none that I've managed to discover. Zippy has plenty of freedom of movement within the boundaries of the maze which is much larger than the visible playing area and slides around unpredictably instant on splattering Zippy.

The purpose of the game is to explore the maze, scoring points by gobbling up peas, plants and terrible thimbleweeds which have a habit of turning the tables on Zippy. There are seven skill levels in all and you have to race against the

timer keeping Zippy out of trouble long enough to reach the next level.

The main obstacle to avoid are the spikes which appear for the first time in level two, the water be-

cause Zippy can't swim and the sides of the playing area. There are also things called absorbers Zippy, which could be good, bad or just plain ugly but my failure to progress beyond level two has failed to reveal them.

The lack of any useful instructions doesn't help matters much either.

The great thing about Splat is that reasonable skill levels are quickly achieved and it has a perfect frustration factor which makes you keep plugging away at the high score.

One thing that might help, but I'm not sure you should rely, is slow motion movement when the pause facility is on. Not only will help calm your nerves but can also help you get through the difficult bits.

S L P



Fancy having a go at
writing your own
arcade-type game?
William Tong has tried
to give you some
guidelines to start you
on the way.

YOUR OWN ARCADE GAME



HOW MANY TIMES HAVE you played a new computer game and after minutes (or sometimes hours!) of frustrated playing you are sure that you could have written a far better game yourself? It is really not as difficult as you might think provided you approach the task in the correct fashion — and you can get a good deal of enjoyment from the process of writing itself, besides when you can happily play with the finished product!

The first move

Naturally your first move is to think of the basic idea for the game. Of course the idea should be an original one — try to break away from the same old mould of alien-bagging, maze-wandering and monster-killing! No doubt this is the most difficult part of writing a good game.

How don't immediately rush off to your computer — well not yet anyway! Transfer your idea to paper — there's nothing worse than spending hours at the keyboard only to find that the game takes at a crucial time — this is a trap to make you give up on the idea before you've given yourself a fighting chance. Think everything out before you get too carried away: what sort of characters will be involved? Are you going to use graphics and sound in a particularly unusual way? What will signify the ending of the game? (Do you want a two-player facility? There are so many other factors will be vitally important to establish before you go

headlong into typing actual code.

Throwing down the gauntlet

First in mind that the game should be challenging to be even the most skilful player. This does not mean that the game should be impossible to beat — people will only lose interest in a game where they haven't got even

the faintest chance of winning! A way of getting round this problem could be the use of different levels of difficulty. Above all keep the concept simple — too many rules can be confusing and they should remain the same throughout the game.

Most people prefer a joystick-operated game but it is wise to keep all your options open. If you have

not got a joystick and are forced to use the keyboard, please give some thought to the choice of controls so that the player does not have to perform super-human digital antics in order to keep a round!

On screen action

Your own arcade game! Having got your ideas and plans on paper, you can



touching the computer. A lot of thought is needed to make the game interesting and addictive — this is often the stage at which programmers don't apply too much care and the small but important touches are left out. In good games the screen is always changing and the graphics are usually exciting. At this stage you can let your imagination run not when thinking of the representation of the main character in the game and here you can use the IBM 64's graphics abilities to the full. But don't worry if you find it difficult — there are many graphics design packages around for the IBM 64.

Colour use is also very important and is probably best added after everything else has been completed. Generally colour should be used to minimise the player's frustration; critical

features and anything that's urgent should stand out clearly, whereas features that are not viewed often should be lightly coloured. Be careful with the colours of adjacent characters as the eye is easily fooled by contrast, try also to avoid filling the screen outside the playing area as this can be distracting and confusing. It is also often worth the trouble to try playing the game on a mono screen — different colours do not always distinguish themselves on a black and white TV.

Listening in

Sound effects should be considered carefully; certainly with the 64's excellent sound effects may be generated. Good use of sound can enhance a game and turn it into something

really special, but it is perhaps one of the most difficult things to perfect. It must suggest impending action to the beginner but not antagonise the advanced player; it is worth leaving in the option of turning the sound off. Trial and error is the only way to perfect different sounds for each character or piece of action — a long, low frequency sound suggests slow movement but you need a short distinct sound for more immediate action.

Try to make your sounds as pleasing as possible and try to avoid shifting from low to high frequencies too abruptly. If you are a little more expert in the musical field you could have a go at writing a little tune to go at the beginning of the game or running throughout. Don't forget though that a catchy little tune heard

redundant trees while you're trying to beat the computer can sometimes make you want to beat the computer instead!

Other factors

Don't forget that throughout the writing of this wonderful game you will need to keep track of where the characters are at any one time — was the alien destroyed and therefore now out of the game, or have you merely blasted it into another package in the maze? There's surely nothing worse than having expertly destroyed a submarine only to find that minutes later it has mysteriously come back to life and suddenly blown your ship to smithereens!

A scoring strategy is purely a matter of personal preference, although it certainly appears that many people like a High-Score record facility. You must also decide whether to have a time limit on the game — will you run out of fuel, money or oxygen?

Game point

Well, that's all there is to it! Now it seems a totally daunting task, but rest assured (!) that the hardest game you will ever have to write will undoubtedly be your first. As you go through the routine for the first time you will gradually learn only little movement routines, exciting explosion effects and more, and often you will find that they fit in better to your Mark II version of the game which has all sorts of refinements added to the original. Throughout the process of writing your own game you must continually test each new stage you add on — who knows how that new routine will affect the part you have already perfected? It is also advisable to always keep back-up copies for those odd occasions when the whole thing crashes — you may not always have the stamina or memory to go back and start at the beginning again!



STUCK?

Try a Course prescribed by Dr Watson



BASIC Courses In Space For The Commodore 64 & VIC 20

An exciting course for all children from 7 to 13. Learn BASIC while enjoying this most exciting space adventure. Master the computer to escape the Alien Spaceship

It combines a story which is compelling reading for the young reader with a carefully structured approach to BASIC intricately interwoven into the text. Also, each BASIC command covered is given a separate, careful explanation in a special 'easy reference' section in the second half of the book.

The tape includes the programs on the ship's computer, and four computer aided learning (CAL) programs which teach the major BASIC commands.

"It should build the interest of an under-13 very well."

— News Computing Weekly



BOOK & TAPE

£9.95

Beginner's BASIC For The Commodore 64

This project-based course guides you step-by-step through BASIC, developing games and utilities using progressively more complex BASIC commands. All the programs developed are provided on tape so that you can get a taste of them first, before working along with the detailed explanations given in the book.

Programs include three video ball games, a character generator utility, a sprite generator utility and a 'compositional' music utility. You also get 'Flonxy Add!', a machine-code utility that adds 28 commands to Commodore 64 BASIC allowing easier programming, sound and graphics.



BOOK & TAPE

£10.50

"The Dr Watson series is rapidly becoming recognised as quality material."

"... great thought put towards... if this was the total content of the course, it would prove very beneficial. However, there is more... much more."

"... the course is relatively straightforward, you in learn more and more by making learning fun."

"... when... manufacturers have been shorted here an introductory manual should be written."

— Commodore Computing International

Beginner's Assembly Language Courses For The Commodore 64, VIC 20 And PET

These courses introduce the real beginner to assembly language programming. No prior knowledge of assembly language is assumed and the aim is to ensure that every reader succeeds. Numerous examples illustrate the points while exercises along with solutions test the understanding.

The tape includes an assembler which assembles and disassembles code anywhere in memory. Also on the tape is a binary/BBC/BBCdecimal CAL tutor program which teaches about the various mathematical notations used in machine code programming.



BOOK & TAPE

£12.50

"There may be easier ways of solving two and two together... But there aren't easier ways of learning how your computer does it."

— PCW

"We are expecting VIC 20 users interested in programming should own this book with care."

"The manual would be worth buying for the reference section alone."

"Don't miss it!"

— Micro News

"I have made more progress in a week than working around for a year with other books and articles."

"Is really super other book."

— The VIC 20/Basic London User Group

"... this book is worth the weight in gold."

"If there was ever a good beginner's guide in this field, then this is it."

— Commodore Computing International

HONEYFOLD

HONEYFOLD SOFTWARE LTD

81 Broadleaf Place

Bath Place

High Street, Basing

London EN5 1ED

Tel: 05-443 4139



Take to the air and fly
with us in Concord.
Fasten your seat belts
tightly though and see
if you can land the
aircraft in William
Fang's simulation for
the 64

CONCORDE II

FOR MANY YEARS the French and British airways have been discussing how to speed up travel and at last they have designed a new Concorde, Concorde II. This should be far better than the original Concorde as it travels much faster and is more economical to run. Being a 'professional' pilot yourself you have been given the chance to have the first test flight. Your route is to Glasgow from Heathrow. The journey is not as easy as it seems as there are quite a few instruments to watch and the level you park will have to depend on the weather conditions you will encounter. The flaps, undercarriage, speed, altitude, deviation radius and artificial horizon are all important instruments to watch.



CONCORDE II

How to fly

When you take off make sure your flaps are down or the plane will not go up! When your flaps are down your speed must not exceed 200mph or you will damage your flaps. Your undercarriage (wheels) must be taken up before the speed reaches 300mph or they will be damaged. However, the undercarriage must be down before the aircraft is below 600ft. If your flaps are up the plane speed must not drop below 210mph.

The distance between the two airports is about 380 miles and because this is a real-time based simulation it will take a long time if you keep the speed at 500mph.

The aim of this test flight is to see how fast you can get to the other airport! Speed appears 1700 mph.

The artificial horizon is an important instrument and it comes in very handy when visibility drops to zero. The deviation readout should be updated often as the pointer is in line with the approaching runway, so try and keep the pointer near zero. The map readout on the British Isles useful as it indicates how far you are from Glasgow (if you know where Glasgow is).

When nearing the runway or any other unknown information the control tower will report to you. When going in to land the speed must be about 280 mph and the altitude less

than 60ft. The flaps must be down and the undercarriage must also be down. When you are above the runway drop your speed to 100mph and altitude to about 20ft then you will get a perfect landing.

Controls

To turn push joystick in port and turn as normal.

To increase altitude pull the joystick back.

To nose dive push the joystick forward and the plane will dive down.

To lower your undercarriage use function key 3. To raise your undercarriage use F1.

To lower flaps use F 6

To raise flaps use F 7

To drop altitude without the

plane doing use "A" to drop speed use "S". To increase speed use the fire button on the joystick.

Watch out!

The runway is only 1,3 miles long so make sure you take off before you reach the end of the runway. When you see the other runway drop your altitude to about 30ft or you will fly over it and all effort would have been wasted.

The game may sound easy to complete but wait until you are in the cockpit. There are levels you choose to increase the difficulty storm, engine failure, strong winds, and many other conditions. Good luck, and happy landings!

```

01 *****
02 *****
03 *****
04 *****
05 *****
06 *****
07 *****
08 *****
09 *****
10 *****
11 *****
12 *****
13 *****
14 *****
15 *****
16 *****
17 *****
18 *****
19 *****
20 *****
21 *****
22 *****
23 *****
24 *****
25 *****
26 *****
27 *****
28 *****
29 *****
30 *****
31 *****
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****

```




Telling deep within
his murky cellar,
Runicaster opens the
portals to Other
Worlds...

TALES FROM THE CRYPT

WHAT IS AN ADVENTURE program? Today we look on adventures as part of our everyday (computer) lives. They come in all shapes and sizes for all the different computers we can buy — where did they start? Many of us today, especially those reading this, own a personal home computer, but a few years ago they did not exist — hard to believe, isn't it? Two or so years ago, the options were severely limited — PCs, TRS-80s, build-your-own, a memory capacity standard of about 8K, some 16K or 32K perhaps, if you were rich. Those were the days when computers and computer talk were still the closely-guarded 'in-tech' secret of just a few people who would go to almost any length to preserve, not knowledge, but bafflement and confusion to those who were not 'signed on the line' members of that close-knit club.

Gone, but...

Fortunately those days are past, not long in time, but dead, dead, dead. The home computer has come of age — young but strong enough to walk on its own! But... see thing those mainstream *Avadis* did have that we can thank them for being unable to keep to themselves was adventure, or more properly, Adventure, for, among the dozen computers of business, medical statistics, research into this and that, was to be found (often illicitly), Adventure. The time was just right for its birth: *Dungeons and Dragons* — was already covering more and more to its banner — a game (or way of life) where you, ordinary mortal, could enter a world so different yet so real, that your life could depend on you knowing the right spell or your being alert at

hearing the very fabric of a simple chest whispering to you that it contained a dozen poison arrows, pointed and ready for action.

Into this environment came Adventure, written (I believe) by Mosen-Crowther and Woods around 1976. What was it? Adventure was a new sort of game (life)

where you could enter an amazing world via your computer terminal — many of them did not even have a VDU. So that meant huddling over some sort of 'teletype' to see the answer to your simple instruction of GO SOUTH. This 'world' within the computer was like no board game where you can see quite clearly

where you will 'land' if you throw a '6'. All you knew was the direction in which you could move — sometimes not even that! What was in that direction, you knew not. You had to explore your surroundings and gradually build up a map of them. I'm sure that one of the most important attractions is the freedom



to move where you choose, not bounded by the time 'walls' imposed by the conventional game.

Back to Adventure. There were many passages to explore, logical puzzles to solve and treasure to find. Adventure taught us like no other game before or since. Research scientists and computer operators, designers and programmers spent (wasted?) hours trying to get past the Troll! Unlike many of the successful arcade games which fade into history after months, Adventure has stood the test of time. Many programmers have tried to copy or improve upon it — a few have succeeded, but only a few.

We have moved to an age that expects pretty

pictures with our adventures; sometimes it is well-done and the pictures play a part in the adventure. Mostly the graphics that some reviewers rave about are only there to enable a product to be 'sold' to the public — some distributors (and much of what is in your local store is determined by the distributors) will not even take adventures unless they have those magic, 'telling' graphics!

Graphics take up memory that could otherwise be used to make the program more interesting and present you with more of a challenge — significantly adding the same to make you feel you are really there! The written word is very powerful, not just by

itself but in combination with your imagination. Alas! You, you other half that sort of mind — or you don't! Many people (your warts) cannot 'get into' Lord of the Rings by J.R.R. Tolkien; those that can are able to populate that world with beings so complete as to be real. The same comments may be made of Larry Niven's science fiction. How many of us have been disappointed at seeing an artist's attempt to reproduce a scene from a book we loved? And an artist has at his disposal a far, far greater resolution than even the best money can buy. Model it in Model 3-D and even hope to stream. Graphics are fine but cannot for the foreseeable future replace the written word.

Adventure used the written word as an author does: descriptions of places and things were full and rich, not short-form; you are in a Cave NSE. As the original was mainstream (and with megabytes of core store, it did take some time before it appeared on home computers) but one way and another, with the sophistication of text compression and the dedication of the programmers, you too can own a version of the first of its type, whatever it may be called! Versions have been around for the Spectrum and hascom for some years. Now we also have them for the BBC and of course our special interest, the CMM 64.

Two versions are produced by software houses that have a good reputation, so it is very much a question of 'you pay your money and take your choice'. The two programs are Colossal Adventure by Level 9 Computing, and Classic Adventure from Melbourne House. Both are based heavily on the original but in the case of Colossal Adventure, the number of locations has been increased by 70 and a completely new end-game has been added! Classic Adventure has a fast LOADING system (Pafloids) which can LOADING time to three minutes; on the reverse sample this

only worked one time in four, so both DO take some time to LOAD! Level 9 is hoping to have a fast LOADING version later in the year.

Both are text-only — but what a text level 9 seems to have someone with a really imaginative mind writing these script! Even so, Classic Adventure is far more descriptive than the average run of text adventures. 'Classic' follows the main-frame more accurately when it comes to the mouse. Whereas I can refer to my much-copied, faded, almost unreadable map, culled from a manuscript version and escape (cheat!) easily from 'Classic', I have to actually work at it in 'Colossal'.

'Classic' has the old spelling mistake and a few anomalies: UP-LOOK GRATE — OK; OPEN GRATE — THERE IS NOTHING HERE WITH A LOCK. This jam a little but does not spoil the overall adventure. GET 'A' AND 'B' — OK, but only 'A' is actually taken — another game allows these multiple directions.

'Colossal' has screen colours that do not entirely suit my television but nevertheless are quite readable.

Choose one. . .

Wherever you choose, do get one of them. So much has been written over the years about Adventure that although you may not make it, you will have subconsciously picked up some threads of the plot. When you start playing you may well occasionally get the feeling that you have been here before.

Adventure games give you the opportunity of exploring another world, place or time. Part of their attraction is the feeling of freedom of decision amplified by our own imaginations. It is said that one picture is worth a thousand words, but when that one picture appears within the framework of an adventure game, it is my opinion that it distorts the one thing that adventure games give us — the total freedom for my imagination to run wild!

Modems seem to be the 'in thing' these days for computer freaks, but how many people know what they can do? Simon Rockman introduces them and their uses to Commodore owners.

INTRODUCING MODEMS

A MODem is ONE of the most exciting black boxes that can be hooked up to a computer. Think of it as a telephone for your computer. The device plugs into the back of a computer using either the edge connector or the cartridge port. On the back of the modem is a lead with the new style telephone jack. This plugs into the BT approved hole in the wall just like a Mackay-Massey telephone. From then on the world is your oyster, but as with a telephone it is no use if you've got no one to ring. There are quite a few major services to dial into.

Main contenders

The major database used by home computer owners is PRISTEL. This covers the whole country, but most people only have to make a local call to log in. To use PRISTEL you have to pay a standing charge of at least five pounds a quarter and this allows you to log into the general area of the system. You are given a ten digit ID and a four digit password, the ten digits are fixed and can only be changed by PRISTELI themselves, the second one is like a combination lock and can be changed by the user. Both of these numbers should be kept secret and the second one should be changed regularly. Pristel provides most of the information found in a general magazine — news, sports and holiday information on a specific topic than

the system provides it may be possible to join a Closed User Group (CUG). These usually cost extra to join and are particularly popular with travel agents, farmers and music owners. The section aimed at music owners is called "PRISTEL Microcomputing" and incorporates the association of computer clubs, Viewline and Microware.

Micronet

Micronet is run like a daily newspaper, it is kept up to date with news daily, often being the best way to keep track of what is happening in the world of computers (apart from buying magazines produced by yours truly) etc. There are special technical enquiry facilities and programs to download, some of them quite good, such as "The Hobbit". At the moment there is very little specifically for the Commodore 64 user. However, with the increasing number of 64 modems available, the force of Commodore owners on Pristel should soon begin to challenge the Spectrum and BBC strongholds. The Micronet section page is very addictive; Pristel is a two-way system and this makes the most of that. The microcomputer section of Micronet is a kind of gossip column, sporting rumors and statistics. Viewline is a less formal version of Micronet offering many similar services but with a "looser" of its own. Access to Pristel Microcomputing costs an extra eight pounds a quarter. One way an information provider (IP) can charge you money for information is by putting a

price on a page. When you look at the charged page the amount in the top right hand corner is added to your bill. This is the way a lot of IP's make their money.

On the whole, Pristel Microcomputing can be used very cheaply — no average bill for Pristel, including the Micromonitoring section plus the add



page charge is around £15 a quarter, about the same sort of price as a daily newspaper.

Into business

There are many business orientated networks; most of these require an eighty column display and so are not really suitable. They also tend to be rather expensive. The main system, which will be available to Commodore 64 owners is British Telecom Gold. This rose to fame when it was broke into on the BBC television program-

me, but claims to be much safer now. Gold is a messaging system: it can be used to send long letters and programs to other users, you can store as much information as you like but pay for the amount of memory this takes up. The system is only really useful if you log in regularly or want to send information that is difficult to read over the phone, like legal documents or programs.

Latent bulletin

You don't have to have a huge mainframe to run a database, there are many

minors doing a similar job. These are called bulletin boards (BBs). They usually run on a fairly computer and are free. Most BBs use a slow rate of data transfer and are not compatible with the system used by Pronet; this means that only modems which are capable of changing the speed at which they operate (known as a baud rate) can talk to these systems. With the increasing popularity of home PCs and more BBs are switching to the system that Pronet uses. The advantage of a BB is that whatever you send is put up instantly, everyone is an information provider. The disadvantage is that only one person can use the system at a time. The popular BBs can be very difficult to get through to, they are usually engaged. BBs are only local and it is necessary to dial the site where the computer is set up.

Getting switched on

Bigger computers usually go through a switching system, the main one is PSS. This covers most of the country and is a kind of motorway for data; you dial a local number on the grid (a model, error or identification number, a password and who you want to talk to) and the system puts you through. This service is pretty cheap, data is charged for per packet (a number of bytes) and is much more economical than calling direct long distance, especially when dealing with other computers. PSS requires the person you want to talk to to also be on the system.

CompuNet

The most exciting database from the Commodore 64 owner's point of view is 'CompuNet' specially set up by Commodore in conjunction with a company called ADP. To use CompuNet it is necessary to buy Commodore's own modem.

Commodore have been spectacularly successful in

the US with their two modems for the VIC and 64. They hope to repeat this in Europe by turning the telephone into the computer's most useful peripheral. There are major differences between the American standards for modems and European ones so it was decided to build a new one from scratch in the UK. The job of designing the hardware and writing the Viewdata software was given to ST in Watford. This has been around for a long while now and has an established user base. With the modem comes special software to allow you to use CompuNet.

When you log into CompuNet for the first time all the extra software that you need is sent to you by the system. This includes a routine to allow you to talk to other users direct. CompuNet is designed as a cross between Pronet and a BB. All user information providers but they can charge for the information they provide. There is a system of menus and you have the ability to communicate directly with other users and there are plans for a multi-player game like the American 'Mega wars' or even 'MUD' space and adventure games which allow you to play against other players in real time. Each Commodore modem is unique; it contains a code number (like the registration on a car) which tells the system who is calling. This adds a great amount of security to the system and it is hoped that holidays, full home-banking and being well will be sold through CompuNet.

Commodore have built 3000 modems, they have the plans for 7000 and are to sell 40,000 by Christmas. One person I spoke to said "We will make Microsoft look silly". These plans are certainly ambitious, other computers will be allowed on after a while (Apple are known to be interested), but it is the Commodore 64 owner who will get first crack of the whip. Watch future issues of Your Commodore for details.



Take those earplugs
out and flex those
fingers — William
Fong has written a
helpful music
program for those
learning to play a
musical instrument.

AMONG MUSIC ON THE Commodore 64 using CommodoreBASIC has long and tiring process. I have written a program which will allow you to play any tune you wish and have it played back at your chosen speed and instrument.

When you RUN the program the screen will display the note list and a summary of the instructions. You do not play the notes as you would on a piano but you have to press the notes'



MAKING MUSIC



names, and when you do the notes will be displayed on the scales in their correct places. I have done it this way because you will not be learning if you pressed any old note; I had to go through this stage whilst learning the piano. For those who do not play any instruments there is a summary of the notes playable:

MIDDLE C-D-E-F-G-A-B-C-D-F

Instructions

[1] This will play back the

music you have just created (CBM+ PLAY KEY) This will play the sharp of the note depressed.

[2] Stops music from continuing play back.

[CLR/HOME] This will clear all the variables giving you a fresh start.

[1-9] This will delete a note (1-9) The numbers from one to nine will play the playback at different speeds, one being the fastest.

[R] This will play a one beat rest and display a rest on the scale.

[F] This will display all the

data needed to produce the tune you have created. This is very useful as writing music for your own program is tedious up

(FUNCTION KEYS) These will choose the different instruments.

(SHIFT WITH HOME) This will give you a note an octave higher, A.B. The highest note is high I.

This is only a very short program for what it does but it will be very helpful for the beginner who's learning to play any instrument.

(When the whole scale is full



Having problems?
Want to mean? Like
to praise? This is your
chance to put pen to
paper and air your
views or state your
case. Just write to us
and see if we can
help.

INPUT

Dear Sir,
 I am a beginner to computers, having bought a Commodore 64, with which I have no complaints. However, I do have a problem with programming. The problem is to do with the symbols which are contained in so many programs for the 64. I find it hard to distinguish the symbols from the printed sheet and I have very little understanding of these symbols and their exact meaning. I would be grateful if you could give me some tips on making the symbols clearer and also the titles of any books which will explain the symbols a little more clearly than the Commodore 64 Handbook. Yours faithfully,
 David Harris
 Redmarches

We answer,
 The symbols can be confusing, especially when getting confused with ASCII codes, screen codes and BASIC codes. To help with this, the best book to buy would be the Programmer's Reference Guide which will help you with all aspects of programming on the C64. The only other method is to persevere and deliberately use the symbols in your programs so that you will forcibly learn them and understand them.

Dear Sir,
 I do not understand how to use machine code on my VIC 20. Please could you show me a small machine code program where I can put letters after every line and type it in at the end in blocks. I use long lists of letters, symbols, numbers and spaces which, when entered, give the message "SOS" (S.O.S.). Yours faithfully,
 Simon Okada
 Cambridge

We answer,
 Here is a short machine code program with a basic loader.

```

100 000
200 000 1 1 and then 10
30 poke 100 poke 100 20
40 100 100 100 100 100
50 data 100 0 0 0 0 0 0
60 100 100 100 100 100 100 100
70 100 100 100 100 100 100 100
80 100 100 100 100 100 100 100
90 100 100 100 100 100 100 100
100 data 100 100 100 100 100 100 100 100
110 data 100 100 100 100 100 100 100 100
120 data 100 100 100 100 100 100 100 100
130 data 100 100 100 100 100 100 100 100
140 data 100 100 100 100 100 100 100 100
150 data 100 100 100 100 100 100 100 100
160 data 100 100 100 100 100 100 100 100
170 data 100 100 100 100 100 100 100 100
180 data 100 100 100 100 100 100 100 100
190 data 100 100 100 100 100 100 100 100
200 data 100 100 100 100 100 100 100 100
210 data 100 100 100 100 100 100 100 100
220 data 100 100 100 100 100 100 100 100
230 data 100 100 100 100 100 100 100 100
240 data 100 100 100 100 100 100 100 100
250 data 100 100 100 100 100 100 100 100
260 data 100 100 100 100 100 100 100 100
270 data 100 100 100 100 100 100 100 100
280 data 100 100 100 100 100 100 100 100
290 data 100 100 100 100 100 100 100 100
300 data 100 100 100 100 100 100 100 100
310 data 100 100 100 100 100 100 100 100
320 data 100 100 100 100 100 100 100 100
330 data 100 100 100 100 100 100 100 100
340 data 100 100 100 100 100 100 100 100
350 data 100 100 100 100 100 100 100 100
360 data 100 100 100 100 100 100 100 100
370 data 100 100 100 100 100 100 100 100
380 data 100 100 100 100 100 100 100 100
390 data 100 100 100 100 100 100 100 100
400 data 100 100 100 100 100 100 100 100
410 data 100 100 100 100 100 100 100 100
420 data 100 100 100 100 100 100 100 100
430 data 100 100 100 100 100 100 100 100
440 data 100 100 100 100 100 100 100 100
450 data 100 100 100 100 100 100 100 100
460 data 100 100 100 100 100 100 100 100
470 data 100 100 100 100 100 100 100 100
480 data 100 100 100 100 100 100 100 100
490 data 100 100 100 100 100 100 100 100
500 data 100 100 100 100 100 100 100 100
510 data 100 100 100 100 100 100 100 100
520 data 100 100 100 100 100 100 100 100
530 data 100 100 100 100 100 100 100 100
540 data 100 100 100 100 100 100 100 100
550 data 100 100 100 100 100 100 100 100
560 data 100 100 100 100 100 100 100 100
570 data 100 100 100 100 100 100 100 100
580 data 100 100 100 100 100 100 100 100
590 data 100 100 100 100 100 100 100 100
600 data 100 100 100 100 100 100 100 100
610 data 100 100 100 100 100 100 100 100
620 data 100 100 100 100 100 100 100 100
630 data 100 100 100 100 100 100 100 100
640 data 100 100 100 100 100 100 100 100
650 data 100 100 100 100 100 100 100 100
660 data 100 100 100 100 100 100 100 100
670 data 100 100 100 100 100 100 100 100
680 data 100 100 100 100 100 100 100 100
690 data 100 100 100 100 100 100 100 100
700 data 100 100 100 100 100 100 100 100
710 data 100 100 100 100 100 100 100 100
720 data 100 100 100 100 100 100 100 100
730 data 100 100 100 100 100 100 100 100
740 data 100 100 100 100 100 100 100 100
750 data 100 100 100 100 100 100 100 100
760 data 100 100 100 100 100 100 100 100
770 data 100 100 100 100 100 100 100 100
780 data 100 100 100 100 100 100 100 100
790 data 100 100 100 100 100 100 100 100
800 data 100 100 100 100 100 100 100 100
810 data 100 100 100 100 100 100 100 100
820 data 100 100 100 100 100 100 100 100
830 data 100 100 100 100 100 100 100 100
840 data 100 100 100 100 100 100 100 100
850 data 100 100 100 100 100 100 100 100
860 data 100 100 100 100 100 100 100 100
870 data 100 100 100 100 100 100 100 100
880 data 100 100 100 100 100 100 100 100
890 data 100 100 100 100 100 100 100 100
900 data 100 100 100 100 100 100 100 100
910 data 100 100 100 100 100 100 100 100
920 data 100 100 100 100 100 100 100 100
930 data 100 100 100 100 100 100 100 100
940 data 100 100 100 100 100 100 100 100
950 data 100 100 100 100 100 100 100 100
960 data 100 100 100 100 100 100 100 100
970 data 100 100 100 100 100 100 100 100
980 data 100 100 100 100 100 100 100 100
990 data 100 100 100 100 100 100 100 100
1000 data 100 100 100 100 100 100 100 100

```

Dear Sir,
 I have a Commodore 101/20 and would like to expand it to 64K memory. I already have two 16K and a 16K RAM Pack and a Stack 4 slot available on my board.
 1) How can I use the second 16K RAM Pack to expand beyond the 128 that I got with one RAM Pack in place?
 2) Do you think that since there are now 16K programs for a 32K VIC that I am wasting my time and money in buying extra, and

I should I sell one of the 16K RAM Packs and buy a 32K RAM Pack (faithfully,
 A Kuri
 Angkor on Thames)

We answer,
 Do not buy the 32K ram pack as you do not need it and also because plug-in extras like adventure cartridges, machine code monitors and so on cannot be used with it. To make full use of your pack, open them up and inside will be banks of switches or solder blocks, depending on how old they are. The pins are numbered from 1 - 21. On one pack, make pins 18 and 21 and on the other pack make pins 12 and 13.

Dear Sir,
 I have a VIC 20 on which I have been writing programs since last June. In the program I have written, I usually use user-defined graphics. Now I have a problem! Last Christmas, I have received a VIC RAM pack which now gives me 16.5K of usable memory. The problem is that I can't use my user-defined graphics in the way that I used to, eg.
 10 POKE 32768 POKE 32769
 20 POKE 32770 TO 32771
 30 READ A POKE 32772 A
 40 POKE 32773 TO 32774
 50 DATA etc
 which allows four characters starting at screen code 32

On the paper which comes with the RAM pack it states that "it is not possible to move the VIC screen or character set into external memory". Can you help me by publishing a program which allows you to declare a

character set on the external memory with a 16K RAM Pack fitted and putting the LOCs on the screen? Yours faithfully,
 Wayne Beauchamp
 Norfolk

We answer,
 If you type the following:
 10 poke 64830 : poke 43,1 : poke 44,0
 20 poke 550 : poke 56,56 : poke 56,404,0
 30 printchr(140) : new

This means that your program will now work and all your existing graphics software will also work.

Dear Sir,
 I have just purchased a C64 64 which I added many happy hours with. However I could not get my hands on a C64 tape recorder, so I bought an Atari tape deck. This works fine on loading programs which I have typed into the computer, but on some software which I have bought such as all Interceptor software and Microelectronics' Vegas Jackpot do not run. On screen it reads OUT OF ADDRESS or PARSED in 30. Is it possible to connect the flash in the tape deck or have I wasted my money? Yours faithfully,
 John Adams
 Brixton

We answer,
 The most likely problem with this is that your recorder has no pause facility. When the computer has finished loading from cassette, it automatically pauses in its protection

OUTPUT

E

You could well drive
yourself completely
round the bend with
this exciting game for
the unexpanded VIC.
By Alan Andrew
Lynx

TRACK KING

ARE YOU SITTING COMFORTABLY? Is your seat belt firmly fastened and your mirrors steady? Then you can begin! Use your skill as a driver to reach the end of the race track and still be alive! You have to choose the correct times to accelerate and decelerate or you could end up crashing into the barriers or into the back of another car.

Key to success

The game is played under the direction of the keyboard. U slows down the car, H moves the car to the left, J moves the car to the right and M increases your speed. By keeping the U key pressed, you will slow down more each time you move and so if you want to

slow down by only the slightest fraction, press L once and then take your finger off the key. The same principle applies while speeding up using the M key.

Every time the level changes, the track becomes longer making it harder for you to reach the end!

The details

The game is made up of three programs. Program 1 consists mainly of PRINT statements although some FOR/NEXTs are involved. Program 2 defines the graphics. Program 3 contains the retro-getting stuff and the variables and line functions are given:

Line explanation

LINE	FUNCTION
5-8	Two player variable line screen
9-12	Small screen in and out
13	Set screen ready for game
14	Print sides of track
15-16	Variables
20-29	Main section of program
40-42	Crash routine
49	Level with Line 47
50-61	Player reaches end: Print what level he is at
1000	Sound for every move
2000	Level for crash graphics
3000-3005	Sound for ambulance when crash occurs

Variable explanation

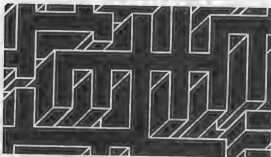
VARIABLE	FUNCTION
SC	Score
H1	Highest score
LC	Level
IC	Finishing distance
TA	Distance travelled
TX	1st car
Y	30720
SP	Speed
SI	30720 sounds, etc
PI	Check for key pressed
L and M	Car that moves towards you
KY	Checks your car





You have to cope with
 mazes (rather than
 snakes) with the
 ladders in this great
 game from Andrew
 Kaycock.

LADDER MAZE



A COMPASS MAY BE OF some help in this game, although I very much doubt it, and I'm not sure that a ball of string would add much to your chances of survival. You must guide your man (could his name

be M) by any chance, through the maze and up the ladders.

The L key moves your man up the ladders, the H key moves him to the left and the J key moves him to the right. Ratings are given

on the time it takes you to reach the top so use your skills to get the best rating you can and see if you can rise to the challenge!

To explain...

Program 1 is an

instructions program made up almost totally of PEEK statements.

Program 2 can be broken down as follows:

Program 2 explanation

LINE	USE	3000-3010	Screen between games.
0	Starts at 1000	4000	Set Y5
1-9	PRINTS maze lines and first ladder. Sets some variables. Main part of game movements, etc.	5000-6000	Ratings
10-22	"You made it". Player reaches top.	The variables used are: A and X	Used to set first screen: X = ladders.
30	Reprints ladders. Chooses new position for ladder.	OP	Used to make man fall
100-110	Send data for graphics.	Z	Set first ladder.
1000-1006	Set screens	M	Man
2000-2004	When player reaches top, send him back down.	T5	Time
		U	Checks for key being pressed
		H	Check if new ladder is needed
		N1	Used as PEEK for man
		M4	Subroutine for T5



1000

```

1  # 1. 데이터 불러오기
2  import pandas as pd
3  import numpy as np
4  import matplotlib.pyplot as plt
5  import seaborn as sns
6  from sklearn.preprocessing import StandardScaler
7  from sklearn.model_selection import train_test_split
8  from sklearn.metrics import r2_score, mean_squared_error
9  from sklearn.linear_model import LinearRegression
10 from sklearn.ensemble import RandomForestRegressor
11 from sklearn.svm import SVR
12 from sklearn.neighbors import KNeighborsRegressor
13 from sklearn.tree import DecisionTreeRegressor
14 from sklearn.metrics import mean_squared_error, r2_score
15
16 # 2. 데이터 불러오기
17 data = pd.read_csv('data/train.csv')
18
19 # 3. 데이터 탐색
20 data.head()
21 data.info()
22 data.describe()
23
24 # 4. 데이터 전처리
25 # missing value 처리
26 data.isnull().sum()
27 data.dropna(inplace=True)
28
29 # feature scaling
30 scaler = StandardScaler()
31 data[['L1', 'L2', 'L3', 'L4', 'L5', 'L6', 'L7', 'L8', 'L9', 'L10', 'L11', 'L12', 'L13', 'L14', 'L15', 'L16', 'L17', 'L18', 'L19', 'L20', 'L21', 'L22', 'L23', 'L24', 'L25', 'L26', 'L27', 'L28', 'L29', 'L30', 'L31', 'L32', 'L33', 'L34', 'L35', 'L36', 'L37', 'L38', 'L39', 'L40', 'L41', 'L42', 'L43', 'L44', 'L45', 'L46', 'L47', 'L48', 'L49', 'L50', 'L51', 'L52', 'L53', 'L54', 'L55', 'L56', 'L57', 'L58', 'L59', 'L60', 'L61', 'L62', 'L63', 'L64', 'L65', 'L66', 'L67', 'L68', 'L69', 'L70', 'L71', 'L72', 'L73', 'L74', 'L75', 'L76', 'L77', 'L78', 'L79', 'L80', 'L81', 'L82', 'L83', 'L84', 'L85', 'L86', 'L87', 'L88', 'L89', 'L90', 'L91', 'L92', 'L93', 'L94', 'L95', 'L96', 'L97', 'L98', 'L99', 'L100']] = scaler.fit_transform(data[['L1', 'L2', 'L3', 'L4', 'L5', 'L6', 'L7', 'L8', 'L9', 'L10', 'L11', 'L12', 'L13', 'L14', 'L15', 'L16', 'L17', 'L18', 'L19', 'L20', 'L21', 'L22', 'L23', 'L24', 'L25', 'L26', 'L27', 'L28', 'L29', 'L30', 'L31', 'L32', 'L33', 'L34', 'L35', 'L36', 'L37', 'L38', 'L39', 'L40', 'L41', 'L42', 'L43', 'L44', 'L45', 'L46', 'L47', 'L48', 'L49', 'L50', 'L51', 'L52', 'L53', 'L54', 'L55', 'L56', 'L57', 'L58', 'L59', 'L60', 'L61', 'L62', 'L63', 'L64', 'L65', 'L66', 'L67', 'L68', 'L69', 'L70', 'L71', 'L72', 'L73', 'L74', 'L75', 'L76', 'L77', 'L78', 'L79', 'L80', 'L81', 'L82', 'L83', 'L84', 'L85', 'L86', 'L87', 'L88', 'L89', 'L90', 'L91', 'L92', 'L93', 'L94', 'L95', 'L96', 'L97', 'L98', 'L99', 'L100']])
28
29 # 5. 모델 학습
30 # Linear Regression
31 X_train, X_test, y_train, y_test = train_test_split(data[['L1', 'L2', 'L3', 'L4', 'L5', 'L6', 'L7', 'L8', 'L9', 'L10', 'L11', 'L12', 'L13', 'L14', 'L15', 'L16', 'L17', 'L18', 'L19', 'L20', 'L21', 'L22', 'L23', 'L24', 'L25', 'L26', 'L27', 'L28', 'L29', 'L30', 'L31', 'L32', 'L33', 'L34', 'L35', 'L36', 'L37', 'L38', 'L39', 'L40', 'L41', 'L42', 'L43', 'L44', 'L45', 'L46', 'L47', 'L48', 'L49', 'L50', 'L51', 'L52', 'L53', 'L54', 'L55', 'L56', 'L57', 'L58', 'L59', 'L60', 'L61', 'L62', 'L63', 'L64', 'L65', 'L66', 'L67', 'L68', 'L69', 'L70', 'L71', 'L72', 'L73', 'L74', 'L75', 'L76', 'L77', 'L78', 'L79', 'L80', 'L81', 'L82', 'L83', 'L84', 'L85', 'L86', 'L87', 'L88', 'L89', 'L90', 'L91', 'L92', 'L93', 'L94', 'L95', 'L96', 'L97', 'L98', 'L99', 'L100']], data['L101'], test_size=0.2, random_state=42)
32 model = LinearRegression()
33 model.fit(X_train, y_train)
34 y_pred = model.predict(X_test)
35 r2_score(y_test, y_pred)
36 mean_squared_error(y_test, y_pred)
37
38 # Random Forest
39 model = RandomForestRegressor()
40 model.fit(X_train, y_train)
41 y_pred = model.predict(X_test)
42 r2_score(y_test, y_pred)
43 mean_squared_error(y_test, y_pred)
44
45 # SVM
46 model = SVR()
47 model.fit(X_train, y_train)
48 y_pred = model.predict(X_test)
49 r2_score(y_test, y_pred)
50 mean_squared_error(y_test, y_pred)
51
52 # KNeighborsRegressor
53 model = KNeighborsRegressor()
54 model.fit(X_train, y_train)
55 y_pred = model.predict(X_test)
56 r2_score(y_test, y_pred)
57 mean_squared_error(y_test, y_pred)
58
59 # Decision Tree
60 model = DecisionTreeRegressor()
61 model.fit(X_train, y_train)
62 y_pred = model.predict(X_test)
63 r2_score(y_test, y_pred)
64 mean_squared_error(y_test, y_pred)
65
66 # 6. 모델 평가
67 # Linear Regression
68 X_train, X_test, y_train, y_test = train_test_split(data[['L1', 'L2', 'L3', 'L4', 'L5', 'L6', 'L7', 'L8', 'L9', 'L10', 'L11', 'L12', 'L13', 'L14', 'L15', 'L16', 'L17', 'L18', 'L19', 'L20', 'L21', 'L22', 'L23', 'L24', 'L25', 'L26', 'L27', 'L28', 'L29', 'L30', 'L31', 'L32', 'L33', 'L34', 'L35', 'L36', 'L37', 'L38', 'L39', 'L40', 'L41', 'L42', 'L43', 'L44', 'L45', 'L46', 'L47', 'L48', 'L49', 'L50', 'L51', 'L52', 'L53', 'L54', 'L55', 'L56', 'L57', 'L58', 'L59', 'L60', 'L61', 'L62', 'L63', 'L64', 'L65', 'L66', 'L67', 'L68', 'L69', 'L70', 'L71', 'L72', 'L73', 'L74', 'L75', 'L76', 'L77', 'L78', 'L79', 'L80', 'L81', 'L82', 'L83', 'L84', 'L85', 'L86', 'L87', 'L88', 'L89', 'L90', 'L91', 'L92', 'L93', 'L94', 'L95', 'L96', 'L97', 'L98', 'L99', 'L100']], data['L101'], test_size=0.2, random_state=42)
69 model = LinearRegression()
70 model.fit(X_train, y_train)
71 y_pred = model.predict(X_test)
72 r2_score(y_test, y_pred)
73 mean_squared_error(y_test, y_pred)
74
75 # Random Forest
76 model = RandomForestRegressor()
77 model.fit(X_train, y_train)
78 y_pred = model.predict(X_test)
79 r2_score(y_test, y_pred)
80 mean_squared_error(y_test, y_pred)
81
82 # SVM
83 model = SVR()
84 model.fit(X_train, y_train)
85 y_pred = model.predict(X_test)
86 r2_score(y_test, y_pred)
87 mean_squared_error(y_test, y_pred)
88
89 # KNeighborsRegressor
90 model = KNeighborsRegressor()
91 model.fit(X_train, y_train)
92 y_pred = model.predict(X_test)
93 r2_score(y_test, y_pred)
94 mean_squared_error(y_test, y_pred)
95
96 # Decision Tree
97 model = DecisionTreeRegressor()
98 model.fit(X_train, y_train)
99 y_pred = model.predict(X_test)
100 r2_score(y_test, y_pred)
101 mean_squared_error(y_test, y_pred)
102
103 # 7. 모델 해석
104 # Linear Regression
105 X_train, X_test, y_train, y_test = train_test_split(data[['L1', 'L2', 'L3', 'L4', 'L5', 'L6', 'L7', 'L8', 'L9', 'L10', 'L11', 'L12', 'L13', 'L14', 'L15', 'L16', 'L17', 'L18', 'L19', 'L20', 'L21', 'L22', 'L23', 'L24', 'L25', 'L26', 'L27', 'L28', 'L29', 'L30', 'L31', 'L32', 'L33', 'L34', 'L35', 'L36', 'L37', 'L38', 'L39', 'L40', 'L41', 'L42', 'L43', 'L44', 'L45', 'L46', 'L47', 'L48', 'L49', 'L50', 'L51', 'L52', 'L53', 'L54', 'L55', 'L56', 'L57', 'L58', 'L59', 'L60', 'L61', 'L62', 'L63', 'L64', 'L65', 'L66', 'L67', 'L68', 'L69', 'L70', 'L71', 'L72', 'L73', 'L74', 'L75', 'L76', 'L77', 'L78', 'L79', 'L80', 'L81', 'L82', 'L83', 'L84', 'L85', 'L86', 'L87', 'L88', 'L89', 'L90', 'L91', 'L92', 'L93', 'L94', 'L95', 'L96', 'L97', 'L98', 'L99', 'L100']], data['L101'], test_size=0.2, random_state=42)
106 model = LinearRegression()
107 model.fit(X_train, y_train)
108 y_pred = model.predict(X_test)
109 r2_score(y_test, y_pred)
110 mean_squared_error(y_test, y_pred)
111
112 # Random Forest
113 model = RandomForestRegressor()
114 model.fit(X_train, y_train)
115 y_pred = model.predict(X_test)
116 r2_score(y_test, y_pred)
117 mean_squared_error(y_test, y_pred)
118
119 # SVM
120 model = SVR()
121 model.fit(X_train, y_train)
122 y_pred = model.predict(X_test)
123 r2_score(y_test, y_pred)
124 mean_squared_error(y_test, y_pred)
125
126 # KNeighborsRegressor
127 model = KNeighborsRegressor()
128 model.fit(X_train, y_train)
129 y_pred = model.predict(X_test)
130 r2_score(y_test, y_pred)
131 mean_squared_error(y_test, y_pred)
132
133 # Decision Tree
134 model = DecisionTreeRegressor()
135 model.fit(X_train, y_train)
136 y_pred = model.predict(X_test)
137 r2_score(y_test, y_pred)
138 mean_squared_error(y_test, y_pred)
139
140 # 8. 모델 배포
141 # Linear Regression
142 X_train, X_test, y_train, y_test = train_test_split(data[['L1', 'L2', 'L3', 'L4', 'L5', 'L6', 'L7', 'L8', 'L9', 'L10', 'L11', 'L12', 'L13', 'L14', 'L15', 'L16', 'L17', 'L18', 'L19', 'L20', 'L21', 'L22', 'L23', 'L24', 'L25', 'L26', 'L27', 'L28', 'L29', 'L30', 'L31', 'L32', 'L33', 'L34', 'L35', 'L36', 'L37', 'L38', 'L39', 'L40', 'L41', 'L4
```

100

[illegible]

[We have a look at
 some of the books
 filling our
 shelves.

REFERENCE LIBRARY

Book Title:
 Secrets of the Com-
 modore 64
Author:
 P. Cornes and A. Cross
Publisher:
 Bernard Babant Ltd.
Price:
 £1.95

THIS POCKET-SIZED manual professes to be yet another beginners' guide to the Commodore 64. Although the book will win no prizes for originality, for those of you who haven't the time or the inclination to plough your way through some of the more weighty volumes on the market, it might prove invaluable as a brief yet informative introduction to the Commodore 64.

The book commences by informing us, for the umpteenth time in the history of computer literature, how the Commodore 64's memory is organised and of the proportion of that memory which is available to the Commodore 64 from BASIC. Some knowledge of the fundamental concepts of the Commodore 64 is obviously presumed since the authors ignore the simpler facets of the BASIC programming language by launching straight into such relative complexities as the all-important random numbers and means of generating them, the IF/THEN and GOTO statements, the GET statement and input routines. The reader is also enlightened on character graphics, sprites and high resolution graphics. Chapters on sound, in the realm of both sound effects and music creation, and machine code programs and the use of machine code statements to extend BASIC, conclude this book.



On the whole, the book is user-friendly with chapters divided into short sections of succinct, jargon-free explanations elucidated by several diagrams and code listings in easy-to-read bold type.

Book Title:
 Brainbusters for the VIC
 20
Author:
 Catherine Ladinski
Publisher:
 Phoenix Publishing
 Associates
Price:
 £5.95

GENEVEVE LADINSKI, AN experienced programmer and technical author with her own software company

specialising in educational programs, has produced this book of twenty-two brainbusters for those VIC 20 users whose idea of fun is to tear their hair out and box their ears while testing their mental agility. Many of the programs even contain an IQ rating at the end of the program, assessing the speed and dexterity with which you solve the problem in hand.

Five brainbusters utilize the machine's music and graphics capabilities. Ms. Ladinski explains her puzzles briefly and intelligibly and the code for the brainbusters is listed in a clear and readable form.

A fairly wide selection of puzzles are offered ranging from fun-of-the-mill

maze games (Word Search) and maze games (Maze Maze and A-Mazing just that side again!) to musical puzzles (Jokes to Jitters — a musical guessing game) and the obligatory space adventure, Close Encounters. The famous Drake Adventure Game, the program used by Ms. Ladinski as her star performer, lets the reader step back in history and into the shoes of Sir Francis Drake as he sailed off in pursuit of the North West Passage. You also have the chance to be on either side of the line, for the volume amongst you, the book often lets Cracker, a simple gaming game computer with police work of an incorrect guess leaves the cop hot on your trail and blazes. Adventure which allows you to live the life of Lardini in the run from the sheriff after robbing a bank, in true western style. Or put on your detective hat for Detective on an Agatha Christie style Who Dunnit.

So whether you see yourself as cop or robber, musician or explorer, this book provides the reader and VIC 20 user with puzzles to tax the brain while stepping into the shoes of the alter ego; the less adventures amongst you can start to Ms. Ladinski's number and word puzzles.

Book Title:
 Commodore 64 Sound &
 Graphics
Author:
 Peter Falconer
Publisher:
 Melbourne House
 Publishers

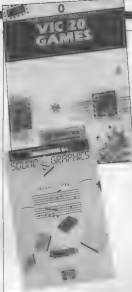
THE COMMODORE 64'S sophisticated sound and graphic capabilities, covered briefly in the



Commodore 64 manual, are incorporated in this book by Peter Falconer. The features available are explained and the user is involved in the design and coding of a real application. The game used for the purpose of illustration throughout the book is written in both BASIC and machine code, thus enabling the user to exploit the full potential of the Commodore's sound and graphics features. But don't fret if you're not to discover the delights of machine code since Mr Falconer presents his machine code routines in three ways: as assembly language code, as BASIC programs illustrating the same idea (where possible) and as DATA statements for inclusion in BASIC programs. The author has squeezed a lot of useful information into a small book and has made this information easy to follow by presenting it in small blocks of text interspersed with examples of code. The book is divided into two sections, the first on low resolution graphics, on character graphics, on bit mapped graphics. As per usual, the user's ability to follow the code is hampered by the illegible Commodore graphic windows.

The book starts with a simple game program written in BASIC. The game is enhanced throughout the book by the addition of graphics, sound and more professional presentation. These features are covered chapter by chapter. The mysteries of sprites, music and interrupts are explained, based on the Commodore's (voices, waveforms, volume, for example) is explored and the more complex facilities of the Commodore 64 such as scrolling, sprites and high resolution graphics are investigated.

Having pored the game together chapter by chapter, the user should be left with the know-how to use the Commodore 64's graphics and sound features when developing his own games and applications.



Book Title:
Business System, on the Commodore 64

Author:
Susan Curran and Margaret Norman
Publisher:
Granada
Price:
£6.95

THIS BOOK, WRITTEN BY Susan Curran, a full-time writer specialising in computers and their applications, and Margaret Norman, a freelance writer and computer programmer, provides a practical and

informative introduction to the Commodore 64 as a small business computer. It shows how the Commodore 64, low priced yet well established and highly reliable is a crucial factor when considering hardware for business applications, can be employed to run business application packages.

The book starts with a general explanation of the Commodore 64, covering why it is a good machine for the small business and painting a broad picture of the business software

available. It does not profess to be a maintenance guide or programming guide nor is it merely a brightly lit list of business applications available for use on the Commodore 64. Each chapter commences with some background information into the type of application under discussion. The text is clearly illustrated with computer printouts and screen displays. The reader is told how to access the manual, accounting, stock-taking, or whatever the business application in question may be, to complement his business. The types of software available for a particular application are then discussed.

The applications included are computerised accounts, stock control and other money-oriented programs, spreadsheets, databases and filing systems, word processing programs and a final chapter on applications included from the list.

To conclude, "Business Systems on the Commodore 64", provides a clear and concise introduction to the different types of business applications which may be run on the Commodore 64 and explains how they may be integrated into the small business.

Book Title:
The Penguin Book of VIC 20 Games
Author:
Paul Copeland
Publisher:
Penguin
Price:
£2.95

PAUL COPELAND'S BOOK of VIC 20 games, at £2.95, is a good value for money. The book contains an assortment of adventures, arcade-type games and board games. The author's musical background is reflected in the sophisticated musical routines in some of the games. Life, Al-Khwarizm and Music Sequence, for example. As well as ten games, the book includes chapters on utility programs and creating your own

business applications for the commodore 64

technology, and a subscription for business users

james hall



games. The author claims to have fully explored the colour, sound and graphics potential of the VIC-20.

The first two games, You Draw and Noughts and Crosses, may be adapted for use on disc-based systems. All the games are introduced by a brief synopsis of what should happen if you enter the program code correctly and by descriptions of the program structure. The game listings conclude the chapters.

The games vary from the commonplace — Noughts and Crosses — to the adventurous such as Red Alert where, as commander-in-chief of a space ship you are sent on a secret mission into outer space to place various space modules in quadrants of space.

The entrepreneurs amongst you may wish to test your skills at Oil Rig whereby, as director of an oil mining company, you have to find as many oil fields as possible (up to a maximum of ten), if your love lies in gambling rather than business, turn back to the Las Vegas of 1942 where your target is to break the bank before it breaks you.

Less philistine readers may wish to take their creative talents to the small screen with You Draw or Water Language.

A set of handy utility routines and a group of remarks to help you create your own games can be found at the end of the book.

Book Title:
Mastering Machine Code on your Commodore 64

Author:
Mark Greenfield
Publisher:
Interface Publications
Price:
£7.95

THIS GLOSSY LITTLE volume, written by the author of "Mastering the Commodore 64", allows the Commodore user, already conversant in BASIC, to create more efficient, faster and professional programs

through the medium of machine code — in this case, the 6502/6510 Assembly language, available for use on the Commodore 64.

The book comprises three sections and numerous informative appendices. Section one commences with a listing and explanation of a 6510 assembler/disassembler/monitor, SUPER-MON. Since all the programs in the book are listed in mnemonic format, an assembler is needed to enter them. This section contains with explanations of every 6510 Assembly language command and every programming mode of the 6510 chip. The tutorial is interspersed with copious and hand examples of Assembly language programs.

The second section shows the user, having now achieved some proficiency in Assembly language programming, how to capitalise on his newly acquired skills by putting them to practical use. The use of Assembly language in

working (both with posix and characters), sprites, music beeps and tones on one and three channels, interrupts, raster scan graphics, high resolution graphics and, finally, adding commands to the BASIC language, are covered. As with the first section, Mr Greenfield's text is clarified by appropriate examples and listings. Furthermore, he invariably couches his explanations in lay-man's terms, thus maintaining his back-cover promise to produce a book for the Assembly language beginner.

The third and final section covers the 8088 routines inside the Commodore 64 and instructs the user on how best to apply them to their programs.

The book ends with a generous spread of appendices (12 in all) many of which appear to have been reprinted from the Commodore 64 manual (ASCII values for characters and control codes).

Book Title:
Business Applications for the Commodore 64
Author:
James Hall
Publisher:
Sunshine Books
Price:
£5.95

FOR THOSE COMMO-
dore 64 users who wish to use their computer as a business machine without turning to the off-the-shelf business applications market, this book provides a useful introduction to designing and writing bespoke business software in Fortran/Basic. Mr Hall reports the amount of time which may be saved in changing from a manual to a computerised system, is deduced from his own experience. Moreover, he states that there is no need to invest in expensive peripherals to run business programs on the Commodore 64 apart from the use of a printer for some applications such as word processing although he does, later on, use the advantage of a disc drive in providing greater memory capacity and faster file access for business applications. The inexperienced programmer need not fret since these programs are accompanied by step-by-step documentation and illustrations where necessary.

The book introduces the reader to business program design with a synopsis of computers and file handling. The applications covered by the book are constructed from a library of subroutines which can then be amended, added to and tailored for the user's individual needs. A useful final chapter allows the user to enter a routine to replace dot matrix graphic symbols with letters and also includes a utility program.

So, if your Commodore 64 is employed as a business machine and you are brave enough to exploit the advantages of creating your own business applications, this attractive book is a welcome investment.

This great video introduces you to the delights of programming games for the VIC in BASIC. Bryan Phillips takes us through the steps.

THIS IS THE FIRST OF A five-part series of BASIC games programming for the VIC-20. The series is primarily intended for newcomers to game programming, but there might well be a few useful tips for seasoned programmers.

Many people are put off writing action games in BASIC because they think it is too slow, and have read too many others that are good game must be written in Machine Code. That's true up to a point. A badly written ball score version of Space Invaders written in BASIC could be nearly painful. Semi-conscious Alamo jerking down the screen being potted off could be one very short line.

as between cups of coffee would not be exactly inspiring. On the other hand a well written Lunar Lander or Shooting Gallery game could rival many of the commercial products. In fact it would be better, because it would be TILK game with your own amazing graphics and sound effects, and that is what it's all about.

In this series I will attempt to show you some of the techniques which you can use to write effective games programs in BASIC, and if you follow it through and use your imagination you should even be able to do something about that 'Source browser' game!

Abstract

Screen display:
This month the emphasis is on the screen display, and this can be one of the most

100

```

00 PRINT " "
10 PRINT "*****"
20 PRINT " "
30 PRINT " "
40 PRINT " "
50 PRINT " "
60 PRINT "*****"
70 PRINT " "
80 PRINT " "
90 PRINT "*****"
100 PRINT " "
110 PRINT " "
120 PRINT "*****"
130 PRINT " "
140 PRINT " "
150 PRINT "*****"
160 PRINT " "
170 PRINT " "
180 PRINT "*****"
190 PRINT " "
200 PRINT "*****"
210 PRINT "*****"
220 PRINT "*****"
230 PRINT "*****"
240 PRINT "*****"
250 PRINT "*****"
260 PRINT "*****"
270 PRINT "*****"
280 PRINT "*****"
290 PRINT "*****"
300 PRINT "*****"
310 PRINT "*****"
320 PRINT "*****"
330 PRINT "*****"
340 PRINT "*****"
350 PRINT "*****"
360 PRINT "*****"
370 PRINT "*****"
380 PRINT "*****"
390 PRINT "*****"
400 PRINT "*****"
410 PRINT "*****"
420 PRINT "*****"
430 PRINT "*****"
440 PRINT "*****"
450 PRINT "*****"
460 PRINT "*****"
470 PRINT "*****"
480 PRINT "*****"
490 PRINT "*****"
500 PRINT "*****"
510 PRINT "*****"
520 PRINT "*****"
530 PRINT "*****"
540 PRINT "*****"
550 PRINT "*****"
560 PRINT "*****"
570 PRINT "*****"
580 PRINT "*****"
590 PRINT "*****"
600 PRINT "*****"
610 PRINT "*****"
620 PRINT "*****"
630 PRINT "*****"
640 PRINT "*****"
650 PRINT "*****"
660 PRINT "*****"
670 PRINT "*****"
680 PRINT "*****"
690 PRINT "*****"
700 PRINT "*****"
710 PRINT "*****"
720 PRINT "*****"
730 PRINT "*****"
740 PRINT "*****"
750 PRINT "*****"
760 PRINT "*****"
770 PRINT "*****"
780 PRINT "*****"
790 PRINT "*****"
800 PRINT "*****"
810 PRINT "*****"
820 PRINT "*****"
830 PRINT "*****"
840 PRINT "*****"
850 PRINT "*****"
860 PRINT "*****"
870 PRINT "*****"
880 PRINT "*****"
890 PRINT "*****"
900 PRINT "*****"
910 PRINT "*****"
920 PRINT "*****"
930 PRINT "*****"
940 PRINT "*****"
950 PRINT "*****"
960 PRINT "*****"
970 PRINT "*****"
980 PRINT "*****"
990 PRINT "*****"

```

adjustable aspects of writing a program. It's a good place to start because space doesn't matter — the more space you use, the more you can up your pile size. The key point is to get it done good, and that takes planning. One way of doing this is to draw it out on paper first. Graph paper is OK, but don't forget your local screen of dots — it'll be rectangular in only problem with using paper is that it's slow, boring and discourages experimentation. Draw out rough something which looks reasonably good, then try it with a pen, and/or immediately off to a hard disk if you're better at using the screen to experiment, and the saving grace is that the rubbery pen can't do it.

the article will help you to do that. My worst feeling at this point that the VIC 20 has an extremely well thought out graphics subsystem. Unfortunately there is a great temptation to forget this and always get your own set of colored graphics well away from the VIC 20. Part 2. But don't forget it even if you use your own graphics. Incorporate your set of standard graphics into your own set. With you use your own set, particularly strive to use a color set. Often an otherwise good program is ruined by messy graphics. This is because some of the characters and control codes are used. You don't mix very well, to the extent that some of the characters come out

1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 26

series from computer to computer and your TV can also make a difference. Play about with the graphics, make some shapes, draw a series — EXPERIMENT!

The program

So now you've got a screen design and want to put it in your program. There are two options in BASIC on the VIC 20, PRINTing or POKing. Let's look at a simple screen display, such as the one in fig. (1).

You can either PRINT it or POKE it onto the screen. PRINTing is the (easier) option as you simply write a number of PRINT statements containing the design. This is shown in Listing 1.

In contrast, POKing a design onto the screen takes a bit more thought. You have to define your screen co-ordinates and use the appropriate POKE formulae. The simplest method is to regard the screen in terms of X,Y co-ordinates and define the bottom left hand corner as 0,0. So POKE a character into the screen via

POKE P1+X-32*Y,CH

and to POKE a colour via

POKE P1+X-32*Y,CL

Where P1=8194
P2=8888

Using 2 gives an example of how the screen design can be POKed onto the screen.

At first sight PRINTing is better as it's much faster,

Wait a minute though, speed doesn't matter here. What about memory? If you look at the line memory displayed you will see that Listing 1 only left 2929 Bytes free, whereas Listing 2 left 3363 Bytes free, a saving of 248 Bytes, and that could be crucial later on. If you remember Listing 2 you will

see that it could be tightened up even further by using multiple statements (loop, GOSUBs etc), and this is shown in Listing 3 where a further 38 Bytes have been saved, a total saving of 375 over Listing 1.

If you're writing game programs on an unexpanded VIC 20, tight programming is essential. Actually the listings don't look so good, the program can be defined to follow and debug, but you leave yourself enough memory over for the game and with any luck will have enough left over for all those hills that make it look professional — High score record, menu options, keyboard/joystick capability and of course the sound effects. As you will see in a future article tight programming also means sparer action, and it really is worth taking the time to plan ahead, and to practice squeezing everything down as far as you can.

Random thoughts

Finally let's have a look at the RND function. Often it can be used to very good effect in screen design and on a multi-level game can be used to add variety in the game progression. You can use it to change character

Listing 1

```
10 PRINT "P"
20 P1=8194:P2=8888
30 POKeP1:GOTO 100
40 FOR P1=8194 TO 8194+304
50 POKeP1+1:GOTO 100
60 POKeP1+2:GOTO 100
70 NEXT P1
80 GOTO 100
90 POKeP1+100:GOTO 100
100 POKeP1+X-32*Y,CL
120 POKeP1+X-32*Y,C
130 NEXT Y
140 NEXT X
150 POKeP1+X-32*Y,C
160 POKeP1+X-32*Y,C
170 POKeP1+X-32*Y,CL
180 POKeP1+X-32*Y,C
190 NEXT Y
200 PRINT "G"
210 GOTO 100
```

Listing 2

```
10 PRINT "P"
20 P1=8194:P2=8888
30 FOR P1=8194 TO 8194+304:INCR P1:GOTO 100
40 FOR P1=8194 TO 8194+304:INCR P1:GOTO 100
50 FOR P1=8194 TO 8194+304:INCR P1:GOTO 100
60 PRINT "G"
70 GOTO 100
100 POKeP1+X-32*Y,CL:POKeP1+X-32*Y,C:NEXT Y
NEXT X
```

Listing 3

```
10 PRINT "P"
20 P1=8194:P2=8888
30 CH=160:GOTO 100
40 FOR P1=8194 TO 8194+304:INCR P1:GOTO 100
50 FOR P1=8194 TO 8194+304:INCR P1:GOTO 100
60 CH=160:GOTO 100
70 GOTO 100
100 POKeP1+X-32*Y,CL:POKeP1+X-32*Y,C:NEXT Y
NEXT X
110 POKeP1+X-32*Y,C:GOTO 100
```


Just as a change from
the usual type of
book review, we
thought we'd show
you exactly the sort of
thing you'll be getting
for your money.

COMMODORE 64 EXPOSED

THIS ARTICLE IS AN extract from the Graphics chapter of Melbourne House's *Commodore 64 Exposed* by Bruce Bayley, which we are reprinting with their kind permission. The should give you, not just a lot of useful information on sprites, but an idea of the high quality of the book. *Commodore 64 Exposed* costs £6.95 in paperback, and contains 192 pages.

Melbourne House are concentrating heavily on the Commodore 64 this autumn. They have just published *Commodore 64 Sound and Graphics*, and at the end of September *Commodore 64 Machine Language for the Absolute Beginner* will be available. Both books will also be priced at £6.95.

Melbourne House books can be found in all good computer bookshops, or they can be ordered directly at Castle Yard House, Castle Yard, Richmond, TW9 6TP.

Sprites

A sprite is a form of user-defined character that is controlled by a powerful video-chip called the 6564. Up to 8 sprites can be displayed at a time automatically. More sprites can be displayed using Basic Interrupt techniques. Sprites have the following advantages over user-defined characters:

1. Pixel by pixel movement in any direction.
2. The 24 by 23 pixel sprite shape can be moved as though it were a single character.



3. Magnification (2X) in both horizontal and vertical directions.

4. Independent high-res/multicolour mode.
5. Selectable sprite to background overlay priority.
6. Sprites to sprite collision detection.

7. Sprites to background collision detection.

A sprite is larger than a character, therefore more data is needed to define the shape of a sprite. A sprite is 24 pixels (3 bytes) wide and 23 pixels high which gives a total of $3 \times 23 = 69$ bytes of data to define the shape of a single sprite. Even though a single sprite is made up of so

much data, the video chip moves the sprite as if it were a single character.

Sprite Pointers

The 64 byte blocks of data that define the shape of each sprite can be placed in any 64 byte multiple of unused memory. In order to tell the video chip where in memory each sprite-shape block is located, eight sprite pointers are provided.

The shape of a sprite may be changed by adjusting the sprite pointer allocated to that sprite to point to a different block of sprite-shape data. Using this

method a single sprite may be assumed by quickly changing the sprite's pointer to switch through a series of shapes provided for that sprite's animation (e.g. an explosion). Switching the pointer rather than switching between sprites leaves the other sprites free for other use.

The sprite pointers are the last 8 bytes of unused screen memory (2048 - 2047). If you reuse screen memory, the pointers will move with it (but not their contents). You must remember when writing up your sprite pointers that the pointer must point to the first byte within the sprite and that the value in the sprite pointer is the actual memory location of the sprite over 4096. Therefore, the following formula applies:

Location = Sprite pointer * 64

Also if you are not using video bank # 0 (default bank) then you must also add bank number * 16384 to the location. If you haven't switched video banks then don't worry.

Two important points to remember when choosing where to put your sprite data in memory are 1. Its location must be a multiple of 64, and 2. check the memory map to make sure that you are only using spare memory.

Turning Sprites On

For a sprite to be displayed to the screen, it must be turned on. The memory location where the video chip gets its information on which sprites should be



turned on and which should be turned off in location \$1268. The 8 bits within byte \$1268 are labeled from right to left 0—7. Therefore, if we load our sprites from 0—7 then we easily determine which sprites should be on and which should be off by the value contained in byte \$1268. The way that the on/off status of each sprite is determined is as follows:

A 1 in the bit corresponding to the sprite determines that the sprite should be displayed (turned on) and a 0 determines that the sprite should not be displayed (turned off).

eg. 7 6 5 4 3 2 1 0
1 1 0 1 0 1 1 = 295
therefore the statement: POKE \$1268, 295 would supply the value chip with the following information: 'sprites 7, 6, 4, 3, 1 and 0 are to be turned on.
Sprites 5 and 2 are to be turned off.

To turn on a single sprite without affecting the others, use the following statement:

```
POKE $1268, PEEK ($1268) OR ($N)
```

where \$N is the sprite number (0—7).

To turn off a single sprite without affecting the others, use the following statement:

```
POKE $1268, PEEK ($1268) AND (255 - 254)
```

Sprite Colour

High resolution (single colour) sprites can be any one of the 16 colours. The colour of each sprite 0—7 should be POKE'd into their respective colour registers, memory location \$1267—\$1268 (see video register map). Each pixel turned on within the sprite will be displayed in the colour determined by the sprite's colour register; each pixel turned off will be displayed in the colour behind the sprite (i.e. it is transparent).

Multicolour Sprites

In multicolour mode, it is possible to have four different colours in each sprite. Though, as with multicoloured characters, multicoloured sprites have

only half the resolution of single coloured sprites (i.e. pixels must be displayed in pairs). The following table gives the colours determined by each bit-pair combination.

Bit pair	Resultant Colour
00	Transparent (no colour)
01	Not transparent (transparent & 0 colour) \$1267
10	Sprite colour register
11	Not transparent (transparent & 1 colour) \$1268

The register that holds information on which sprites are multicoloured and which sprites are not is mapped to location \$1276.

To set a sprite to multicolour, use the following statement:
POKE \$1276, PEEK (\$1276) OR (\$N4)
where \$N4 is the sprite number (0—7).

To switch a sprite out of multicolour mode, use the following statement:
POKE \$1276, PEEK (\$1276) AND (\$255 - 254)

Expanding Sprites

Sprites can be expanded vertically, horizontally or both. A sprite is expanded by putting 1 pixel in place of 1 and 2 blank pixels of 1 in the direction of expansion thus giving a 2x expansion. To expand a sprite horizontally, the corresponding bit in location \$1277 must be set to 1. To reduce the sprite, the bit must be set to 0. Vertical expansion is done in the same way using location \$1277. The POKE's statements to control expansion and reduction of sprites are as follows:

1 Horizontal expansion:

```
POKE $1277, PEEK ($1277) OR ($N4)
```

OR (\$N4)

Horizontal reduction:

```
POKE $1277, PEEK ($1277) AND (255 - 254)
```

Vertical expansion:

```
POKE $1276, PEEK ($1276) OR ($N4)
```

OR (\$N4)

Vertical reduction:

```
POKE $1276, PEEK ($1276) AND (255 - 254)
```

where \$N4 is the sprite number from 0—7.

Sprite Movement

Sprites are moved around the display by changing the values in each sprite's horizontal and vertical position registers. These

registers are mapped to memory location \$1248 to \$1261 and a most-significant-bit (MSB) register at location \$1266. The MSB register is used to notify the problem of horizontal screen wrap.

The MSB register works as follows. In order to gain pixel movement, the horizontal position register needs to be able to hold values from 0 to 256 (screen width). A single register can only hold values from 0 to 255 therefore we need at least one more bit to handle values up to 256. An extra bit (256 bit) would allow us control over positions 0 to 512. This is the purpose of the MSB register. The bit in the MSB register corresponds to the sprite number: bit 0 for sprite 0, bit 1 for sprite 1, etc. A register map of all sprite positioning registers is given overleaf. Note that horizontal positions 24 and 344 are the left and right boundaries of the screen. Sprites continue to move outside this range but cannot be seen.

It's about time we had a look at one of these sprites.

Study the first program and its comments. Type it in and run it.

Run the program; you should see a square sprite float across the screen.

To expand the sprite in the horizontal and vertical directions before moving, add the following line:

```
65 POKE VIDEO = 25, 1 : POKE VIDEO + 21, 1
```

and run the program again.

The second program allows you to use the cursor keys to draw a sprite by editing DATA statements. Type RUN 1, then use the cursor keys to move around

the DATA statements. Use the shift-Q character to signify a pixel-ON and a tab character to signify a pixel-OFF. When you have finished drawing your sprite, move the cursor to the top of the screen, then keep hitting the RETURN key until you have entered all of the DATA statements.

New type RUN, and the program will process the sprites and the DATA statements needed to generate that sprite. To move these DATA statements, use the same method as you used on the set of DATA statements.

Sprite Display Priorities

Sprite priority determines if the sprite should appear in front or behind another background if the background is another sprite, then the priority is fixed by the sprite's sprite number. Sprite 8 has the highest priority; sprite 1 has the next priority, and so on, up to sprite 7. For example, if sprite 0 and sprite 7 are positioned so that they cross each other, sprite 0 will be in front of sprite 7, though you would be able to see sprite 7 through sprite 0 (unless of course sprite 0 was a completely filled square). Sprites to background priority is more flexible in the way that each sprite can be set with priority above or below the background. The sprite to background priorities are controlled by the sprite priority register (memory location \$1275). A 1 in the bit number corresponding to the sprite number will set that sprite with a lower priority than the background, a 0 in this bit position will give the sprite a higher priority than the background. By moving sprites back and forth over other objects, at the same time changing the sprite-background priorities, it is possible to make it look as if the sprites are moving in front and behind the object thus creating a three dimensional effect.

The third program overlays 8 sprites to demonstrate sprite priority.

Sprite Collisions

Sprite collisions are detected by the computer and collision information is stored in local on 5528 for sprite-to-sprite collisions and location 5529 for sprite-to-background collisions. The bit set to 1 in each of these registers corresponds to the sprite involved in the collision. The bit stays set until the register is read. (I think.) So if the collision information is to be used more than once per collision, it would be a good idea to store the

value into a variable. Also, programs that use the sprite collision registers should realize in their initialization a PEEK of each of these registers to clear them of previous collision data. For example, for par 01 in a multi-colored mode will not be detected as a sprite-to-background collision, even though it can be seen on the screen. So, for example, if you wish to have objects that should not cause a collision (e.g. a cloud) then they should be colored by using a bit par 01 and collision register # 05.

Location	Bit set if
5528	Sprite 0 hit pixel on
5529	Sprite 1 hit pixel on
552A	Sprite 2 hit pixel on
552B	Sprite 3 hit pixel on
552C	Sprite 4 hit pixel on
552D	Sprite 5 hit pixel on
552E	Sprite 6 hit pixel on
552F	Sprite 7 hit pixel on
5530	Sprite 8 hit pixel on
5531	Sprite 9 hit pixel on
5532	Sprite 10 hit pixel on
5533	Sprite 11 hit pixel on
5534	Sprite 12 hit pixel on
5535	Sprite 13 hit pixel on
5536	Sprite 14 hit pixel on
5537	Sprite 15 hit pixel on
5538	Sprite 16 hit pixel on
5539	Sprite 17 hit pixel on
553A	Sprite 18 hit pixel on
553B	Sprite 19 hit pixel on
553C	Sprite 20 hit pixel on
553D	Sprite 21 hit pixel on
553E	Sprite 22 hit pixel on
553F	Sprite 23 hit pixel on
5540	Sprite 24 hit pixel on
5541	Sprite 25 hit pixel on
5542	Sprite 26 hit pixel on
5543	Sprite 27 hit pixel on
5544	Sprite 28 hit pixel on
5545	Sprite 29 hit pixel on
5546	Sprite 30 hit pixel on
5547	Sprite 31 hit pixel on
5548	Sprite 32 hit pixel on
5549	Sprite 33 hit pixel on
554A	Sprite 34 hit pixel on
554B	Sprite 35 hit pixel on
554C	Sprite 36 hit pixel on
554D	Sprite 37 hit pixel on
554E	Sprite 38 hit pixel on
554F	Sprite 39 hit pixel on
5550	Sprite 40 hit pixel on
5551	Sprite 41 hit pixel on
5552	Sprite 42 hit pixel on
5553	Sprite 43 hit pixel on
5554	Sprite 44 hit pixel on
5555	Sprite 45 hit pixel on
5556	Sprite 46 hit pixel on
5557	Sprite 47 hit pixel on
5558	Sprite 48 hit pixel on
5559	Sprite 49 hit pixel on
555A	Sprite 50 hit pixel on
555B	Sprite 51 hit pixel on
555C	Sprite 52 hit pixel on
555D	Sprite 53 hit pixel on
555E	Sprite 54 hit pixel on
555F	Sprite 55 hit pixel on
5560	Sprite 56 hit pixel on
5561	Sprite 57 hit pixel on
5562	Sprite 58 hit pixel on
5563	Sprite 59 hit pixel on
5564	Sprite 60 hit pixel on
5565	Sprite 61 hit pixel on
5566	Sprite 62 hit pixel on
5567	Sprite 63 hit pixel on
5568	Sprite 64 hit pixel on
5569	Sprite 65 hit pixel on
556A	Sprite 66 hit pixel on
556B	Sprite 67 hit pixel on
556C	Sprite 68 hit pixel on
556D	Sprite 69 hit pixel on
556E	Sprite 70 hit pixel on
556F	Sprite 71 hit pixel on
5570	Sprite 72 hit pixel on
5571	Sprite 73 hit pixel on
5572	Sprite 74 hit pixel on
5573	Sprite 75 hit pixel on
5574	Sprite 76 hit pixel on
5575	Sprite 77 hit pixel on
5576	Sprite 78 hit pixel on
5577	Sprite 79 hit pixel on
5578	Sprite 80 hit pixel on
5579	Sprite 81 hit pixel on
557A	Sprite 82 hit pixel on
557B	Sprite 83 hit pixel on
557C	Sprite 84 hit pixel on
557D	Sprite 85 hit pixel on
557E	Sprite 86 hit pixel on
557F	Sprite 87 hit pixel on
5580	Sprite 88 hit pixel on
5581	Sprite 89 hit pixel on
5582	Sprite 90 hit pixel on
5583	Sprite 91 hit pixel on
5584	Sprite 92 hit pixel on
5585	Sprite 93 hit pixel on
5586	Sprite 94 hit pixel on
5587	Sprite 95 hit pixel on
5588	Sprite 96 hit pixel on
5589	Sprite 97 hit pixel on
558A	Sprite 98 hit pixel on
558B	Sprite 99 hit pixel on
558C	Sprite 100 hit pixel on
558D	Sprite 101 hit pixel on
558E	Sprite 102 hit pixel on
558F	Sprite 103 hit pixel on
5590	Sprite 104 hit pixel on
5591	Sprite 105 hit pixel on
5592	Sprite 106 hit pixel on
5593	Sprite 107 hit pixel on
5594	Sprite 108 hit pixel on
5595	Sprite 109 hit pixel on
5596	Sprite 110 hit pixel on
5597	Sprite 111 hit pixel on
5598	Sprite 112 hit pixel on
5599	Sprite 113 hit pixel on
559A	Sprite 114 hit pixel on
559B	Sprite 115 hit pixel on
559C	Sprite 116 hit pixel on
559D	Sprite 117 hit pixel on
559E	Sprite 118 hit pixel on
559F	Sprite 119 hit pixel on
55A0	Sprite 120 hit pixel on
55A1	Sprite 121 hit pixel on
55A2	Sprite 122 hit pixel on
55A3	Sprite 123 hit pixel on
55A4	Sprite 124 hit pixel on
55A5	Sprite 125 hit pixel on
55A6	Sprite 126 hit pixel on
55A7	Sprite 127 hit pixel on
55A8	Sprite 128 hit pixel on
55A9	Sprite 129 hit pixel on
55AA	Sprite 130 hit pixel on
55AB	Sprite 131 hit pixel on
55AC	Sprite 132 hit pixel on
55AD	Sprite 133 hit pixel on
55AE	Sprite 134 hit pixel on
55AF	Sprite 135 hit pixel on
55B0	Sprite 136 hit pixel on
55B1	Sprite 137 hit pixel on
55B2	Sprite 138 hit pixel on
55B3	Sprite 139 hit pixel on
55B4	Sprite 140 hit pixel on
55B5	Sprite 141 hit pixel on
55B6	Sprite 142 hit pixel on
55B7	Sprite 143 hit pixel on
55B8	Sprite 144 hit pixel on
55B9	Sprite 145 hit pixel on
55BA	Sprite 146 hit pixel on
55BB	Sprite 147 hit pixel on
55BC	Sprite 148 hit pixel on
55BD	Sprite 149 hit pixel on
55BE	Sprite 150 hit pixel on
55BF	Sprite 151 hit pixel on
55C0	Sprite 152 hit pixel on
55C1	Sprite 153 hit pixel on
55C2	Sprite 154 hit pixel on
55C3	Sprite 155 hit pixel on
55C4	Sprite 156 hit pixel on
55C5	Sprite 157 hit pixel on
55C6	Sprite 158 hit pixel on
55C7	Sprite 159 hit pixel on
55C8	Sprite 160 hit pixel on
55C9	Sprite 161 hit pixel on
55CA	Sprite 162 hit pixel on
55CB	Sprite 163 hit pixel on
55CC	Sprite 164 hit pixel on
55CD	Sprite 165 hit pixel on
55CE	Sprite 166 hit pixel on
55CF	Sprite 167 hit pixel on
55D0	Sprite 168 hit pixel on
55D1	Sprite 169 hit pixel on
55D2	Sprite 170 hit pixel on
55D3	Sprite 171 hit pixel on
55D4	Sprite 172 hit pixel on
55D5	Sprite 173 hit pixel on
55D6	Sprite 174 hit pixel on
55D7	Sprite 175 hit pixel on
55D8	Sprite 176 hit pixel on
55D9	Sprite 177 hit pixel on
55DA	Sprite 178 hit pixel on
55DB	Sprite 179 hit pixel on
55DC	Sprite 180 hit pixel on
55DD	Sprite 181 hit pixel on
55DE	Sprite 182 hit pixel on
55DF	Sprite 183 hit pixel on
55E0	Sprite 184 hit pixel on
55E1	Sprite 185 hit pixel on
55E2	Sprite 186 hit pixel on
55E3	Sprite 187 hit pixel on
55E4	Sprite 188 hit pixel on
55E5	Sprite 189 hit pixel on
55E6	Sprite 190 hit pixel on
55E7	Sprite 191 hit pixel on
55E8	Sprite 192 hit pixel on
55E9	Sprite 193 hit pixel on
55EA	Sprite 194 hit pixel on
55EB	Sprite 195 hit pixel on
55EC	Sprite 196 hit pixel on
55ED	Sprite 197 hit pixel on
55EE	Sprite 198 hit pixel on
55EF	Sprite 199 hit pixel on
55F0	Sprite 200 hit pixel on
55F1	Sprite 201 hit pixel on
55F2	Sprite 202 hit pixel on
55F3	Sprite 203 hit pixel on
55F4	Sprite 204 hit pixel on
55F5	Sprite 205 hit pixel on
55F6	Sprite 206 hit pixel on
55F7	Sprite 207 hit pixel on
55F8	Sprite 208 hit pixel on
55F9	Sprite 209 hit pixel on
55FA	Sprite 210 hit pixel on
55FB	Sprite 211 hit pixel on
55FC	Sprite 212 hit pixel on
55FD	Sprite 213 hit pixel on
55FE	Sprite 214 hit pixel on
55FF	Sprite 215 hit pixel on

VIC-II Register Map

Reg.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999
Reg.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											

Your

INTEREST MAGAZINE

COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

SO YOU OWN A COMMODORE?

SO YOU'VE WRITTEN SOME PROGRAMS?

SO WHY HAVEN'T YOU SUBMITTED THEM TO US?

Your Commodore is always on the lookout for new material for publication and we know that there are thousands of intelligent, creative, imaginative and creative Commodore owners out there, so why don't we get together?

If you have written an exciting game or an invaluable utility, or your Commodore needs, share your talents with us and our readers by submitting your efforts and the form to the address below. All articles should be double-spaced and typewritten and should be accompanied by a printout of the program as well as a copy of the program on cassette or disc. All material should be original, if it is not chosen for

publication, it will be returned to you.

You may not have written any software yourself, but you have very firm opinions about the world of Commodore and all their attendant industries and products. Then put your opinions on paper and post them to us, again at the address below - you never know, you might even get paid for airing your views! All submissions should be sent to the Editor:

Your Commodore
Angus Specialist Publications Limited
No 1 Golden Square
London W1R 3AB

PLEASE COMPLETE IN BLOCK CAPITALS



Your Name _____

Program Name _____

Computer/memory size it runs on _____

Amount of memory program occupies _____

Other computers/memory size which your program runs on or those envisaged or use _____

Does your game need or use peripherals? _____

Yes

No

Have you sent our game to another magazine? _____

Yes

No

Is it envisaged a version on a disc? _____

Your Address _____

Telephone Number _____

Notes to contact you _____



AN ARGUS PUBLICATION

64 TAPE COMPUTING

For all CBM 64 Users

● 3 Original Games

Archie, Adventure and Strategy — more fun, more value — only from Argus

● UTILITIES

Generators delight a priori — improved graphics, sounds, demonstrations — it's all here!

● On screen reviews of newly launched games

News and views of the wonderful, whacky world of home computing



**NO RISK
GUARANTEE**
Change your mind
No charge

64 Tape Computing adds a new dimension to your micro!

Run this Argus 64 tape and you'll soon see why it's Britain's top selling tape magazine. Each issue gives you a variety of exciting and challenging games to play, reviews of other newly released software plus valuable utilities enabling you to write your own programmes and games.



Stretch your imagination and skills with 64 Tape Computing — available every other month from WH Smith, Mensies and other leading stores.

(You'll see them advertised on TV from September)

Get your copy today!

Argus Tape Magazines produced by
ARGUS PRESS SOFTWARE
1 Golden Square, London W1R 5AB
Telephone: 01 402 6066



Dave Crisp helps out in a retail outlet and he helps here with relative files, which seem to cause a lot of bother.

RELATIVE FILES

JUDGING BY THE AMOUNT of telephone calls, enquiries and questions I receive it would seem that one thing which causes a lot of problems to owners of the 1940 disc drive is relative files. Except for a few things, the manual is more or less useless. Using these two programs, you should at least be able to get BASIC relative files working and once you have got the idea it is easy to start to manipulate files and file sizes etc.

Relatively speaking...

Think of a relative file as a large array. Instead of the array being in RAM though it is stored in numbered order on the disc. As with an array it is easy to find a particular record if you know the number. If the number is not known it is possible to search the disc record by record.

The two programs are very basic. They have simple screen displays, there is very

little error checking, and virtually no string handling. Once you have got the idea of what is going on then you can go back to the manual, which should then make more sense and tidy up the programs.

When creating the file for the first time, I recommend that you use a blank disc. This will ensure that all record numbers are available. Initially create a file of 254 characters/record and have about 100 records. It is worth noting that it is

Program 1. Listing

READY.

```

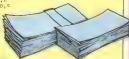
1 DEMONSTRATION PROGRAM 19400000
2 PRINT "q"
3 INPUT "*****HOW MANY CHARACTERS/RECORDS?";RC
4 INPUT "*****HOW MANY RECORDS MAX,500?";RM
10 OPEN "1",B,1:"OPEN2",B,2,"OPEN1 FILE",L,"+CHR$(RC)
20 FOR N=1 TO RM
30 NEXT N
40 PRINT "1940/254/1000-INT 19400000"
45 PRINT "N = " ; RM ; "RL = " ; RC ; " "
50 PRINT "1", "P" ; CHR$(2) ; CHR$(RC) ; CHR$(RM) ; CHR$(10)
55 PRINT "2", "P" ; "S" ; "T"
60 INPUT "1", A, B, C, D, E
70 PRINT "N, A, B, C, D, E"
800 NEXT
900 PRINT "1940/254/10000"

```

READY.

SEARCHING FOR RECORDS
LOADING

READY.



QUICKSILVA'D!



QUICKSILVA

All items available from Quicksilver Staff Order P.O. Box 6, Wimbome Dorset BH21 1PE.

SALESMAN



By John Grisham
Illustrated by
John Grisham

THE
SILVER
SILVER



By John Grisham
Illustrated by
John Grisham

COMES



By John Grisham
Illustrated by
John Grisham

THE
SILVER
SILVER



By John Grisham
Illustrated by
John Grisham

THE
SILVER
SILVER



By John Grisham
Illustrated by
John Grisham

THE
SILVER
SILVER



By John Grisham
Illustrated by
John Grisham

THE
SILVER
SILVER



By John Grisham
Illustrated by
John Grisham

THE
SILVER
SILVER



By John Grisham
Illustrated by
John Grisham

GED
AVAILABLE
SOON



WIMBOME



WIMBOME

WIMBOME

WIMBOME



DATA STATEMENTS

Great reading

TWO WELL-KNOWN AND established book publishers Corp and Addison-Wesley, have joined together to produce a series of computer books and in October they are due to release three books for Commodore owners; all written by Scott Hale, the firm's Arcade Games For Your VIC 20, Arcade Games For Your Commodore 64 and More Arcade Games for Your Commodore 64 to keep your eyes on those book shelves!

Buying Power

Purchase ledger is a new business program available for the Commodore 64 from Kemp Limited. It is a high quality ledger accounting program for recording and analysing purchases and includes normal analysis and will account for VAT. Ten menu options are available; the program can handle up to 100 creditor accounts and up to 90 normal accounts and can deal with up to 660 entries each month. It has a full range of prompts and can also be used with Commodore interfaces. It is disc and tape compatible and can be transferred to disc without difficulty. The retail price is £27.95 on cassette and £29.95 on disc and comes with a 16 page manual. The program can be obtained from good computer shops or from Kemp Limited by mail order. The price includes VAT and postage within Europe; further information can be obtained by contacting Kemp Limited at Millwall Hill, London N16 6PN, tel: 01-444-5499.

New stuff from PSS

MIDWAY, FOR THE Commodore 64, is a diversion from other PSS products, being a strategy war game based on the World War Two clash between Japan and the U.S. The program has been written by a 'war-gamer', who has ensured that the Japanese strategy, (the player takes the role of the U.S.), is the same as that in the real battle. There are three levels, making it suitable for beginners or experts. Midway will be available on disc or cassette at £12.95 and 19.95 respectively.

Commodore 16

THE NEW COMMODORE 16 is the ideal introduction to home computing, offering a powerful 128K RAM, full-size professional keyboard, 121 colours for high-quality graphics and sophisticated sound capabilities. Designed with the first-time user in mind, the Commodore 16 will be sold as a complete 'Starter Pack' comprising: computer, cassette unit, introduction to BASIC Part 1 and four recreational programs — everything needed by the beginner — for £129.95. "The Commodore 16 is the perfect point of entry for anyone interested in serious home computing," said Howard Standish, UK General Manager of Commodore Business Machines. "We've packaged it in a Starter Pack because the VIC-20 has already proved that there's what the public wants, and we believe that in the form it offers the best deal

Getting Stacked

STACK COMPUTERS HAVE dropped the price of their programmers and cartridge for the Commodore 64. Their cartridges provide everything from extensions to BASIC, through fast tape operation and a two pass assembler, range in price from £25.00 upwards. When combined with their unique 4 slot motherboard, which allows use of programmes and games cartridges without powering down, Stack give the standard features of the

64 a very powerful boost.

For those of us with smaller budgets, the Stack 180 range now includes a cassette based compiler. Priced at £14.95 inc. VAT, the end-user can now achieve most professional results with his own BASIC program.

More details are available from Stack Computer Services Ltd, 299-298 Derby Road, Bootle, Liverpool L20 6JH.



Spitely, spitely

GO-SPRITE IS A VERSATILE, non-proprietary screen editor which makes full use of the extensive facilities on the popular Commodore 64, available from Microsoft.

Among the many features of this Microsoft machine driver is the fact that all program features can be operated by joystick alone. The Program also features light pen and keyboard control options.

Go-Sprite gives the user ease of use with iron drawn commands — on-screen symbols give one-touch commands for all program features, simple animation of up to 32 sprites with 255 lines, easy handling of multi-coloured sprites, overlays of up to seven layers, user definable keyboard, sprite data files on disc and tape with data display option, and sprite index.

Go-Sprite was written for Microsoft by Bright Green Software.

anywhere for programming, education or games. The new model features 128K RAM, advanced BASIC 3.5 with over 73 commands, including full graphics plotting and program editing. A unique HELP key is a particularly useful asset when learning to program, as it highlights errors in lines of programming right down to individual instructions. With 128K RAM standard, the Commodore 16 offers a full 128K RAM to the user for BASIC programs, making a powerful hardware for the games enthusiast on the one hand and the programmer or educational user on the other. The computer is equipped with: two joystick ports to permit use of the new advanced Commodore joystick; ROM cartridge and parallel disk drive port; Commodore 1515 Cassette Link interface port; and a Commodore serial port offering access to a wide range of Commodore peripherals.

DATA STATEMENTS

Great Eight

ACTIVISION IS NOW producing games cassettes for the Commodore 64 home computer. Five titles off the production line are *Beamrings*, *Demolition*, *HERO*, *Zero*, *Toy Runway*, *River Raid*, *Refit II* and *Refit II: ON their side*. *Beamrings*, *Demolition*, *River Raid* and *Refit II* are new versions, especially adapted and enhanced for the Commodore 64, of Activision's top-selling 1980

titles. *Refit II* and *HERO* are Commodore-64 versions of Spring 1984 releases which are already high in the charts, and *Zero* and *Toy Runway* are brand new Summer 1984 titles which are being launched simultaneously for the Commodore 64 and other systems.

Zero is claimed to introduce the software entertainment industry's first mystical game of initiation. Begin at the source and feel your way through the unknown, connecting path after path



and you've mastered the mystery and your score. *Zero*, designed for Activision by Matthew Hubbard.

In *Toy Runway*, it's midnight in the toy factory. *Peaceful*. Quiet. All of a sudden the balloon valves open. Gangs of mechanical toys begin bounding from level to level taking over the

toy shop. Can you stop them before they capture you?

Toy Runway, designed by Activision by Mark Turnell. All three new Activision cassettes for the Commodore 64 retail at £3.99 including VAT and are available from video games and home computer software outlets everywhere.

Cartek rides again

FOLLOWING THE SUCCESS of *Cartek & the Microchips*, Commodore have awarded an educational package for the 64, *Cartek & the Kryptobytes*. In the new program, space hero Cartek, together with his Microchips, again demonstrates a new and unusual way of learning to program in Commodore BASIC.

The first program has already been translated into at least five languages and *Kryptobytes* is designed by the same team of three schoolteachers from Southern England. It is accompanied by a full colour story book which contains a 'light file' to teach educational programming skills and programs to type into the computer. An accurate sheet 'map' is used as an overlay in the book to help the student create his own designs and make it easier to POKE them onto the screen. The scenario of the story centres on the planet Xena where the Kryptobytes have established a data centre to store information they have

acquired from other planets. However, a mysterious force begins to rob the information. Cartek and his Microchips are called to the rescue and, with you, have to try to neutralise the force before the safety of the whole universe is jeopardised! Cartek and the Kryptobytes teach students the construction of programs using the REM and LIST commands. The storage of data and the use of the READ command; the application of FOR and NEXT, both as a delay and a nested loop for loops. In addition practice is given in simple program editing; the use of GET and INPUT, as well as a host of other programming features which take the student to an advanced level.

Cartek and the Kryptobytes designed for 10 to 13 year olds, although it is suitable for younger children with parental assistance, while older children and even parents will find it informative and great fun to use.

Pets can microwave

COMMODORE PET USERS can now communicate with the Microwave — the portable hand-held word processor with a unique and extremely simple to use keyboard of just six keys. Microcomputer Services, an appointed Microwriting Centre, has developed the software program 'Speedway', which allows two-way transfer of text between PEs and Microwriter.

Now PET users can transfer text to their data disc for storage, moving of files or for printing out a convenient moment. Documents can also be moved from the PET and stored into the Microwriter's memory for reference, updating or amendment. The Microwriter can also be used in a networked environment.

An interface lead, enabling communication between the PET, which has IEEE connectors, and the Microwriter's in-built RS232C is available from Microcomputer Services.

Due to Microwriter's total portability, it is

specially suited for any situation where text needs to be written away from the office — whilst travelling or for meeting notes, or even at home out of office hours. Within the office, Microwriting provides an invaluable method of producing written text without recourse to dictation. Further documents need to be handwritten and later transferred by a type. The Microwriter user has complete control over content, format and amendments, with materials being written whenever and wherever convenient. Hardcopy printout can also be obtained directly from the Microwriter linked to a printer.

The 'Speedway' program is available from Microcomputer Services, priced at £140 for two-way communications, and £28 for one way. The 'link' will also work with all Commodore models, and details are available from: Microcomputer Ltd., 31 Northampton Row, London WC1. Tel: 01-251-8880.



Super, Supersoft

SUPERSOFT PRODUCES AN enormous range of programs and accessories for the Commodore series, for example...

Available for the C64 is a truly 3-dimensional spread sheet program, **Bussulo 3**. This offers a variety of features over those included in the rest of the spreadsheet range, including up to 999 rows and 255 columns, 3-dimensional formulas and sub-defined functions, and as ribbon sheets, fast F40 command, unique V/L window function, link to Easy Script or VisiCalc and also to Chartpak 64, simple bar charts, sample files and editors and variable-column width. Illustrative columns with all C64 pointers, available as disc only and costs £75 including VAT.

Still on the screen side of computing, Supersoft have taken on the marketing of **Matrix Applications** series, Master, for the C64. They have cut the price to just £69 including VAT, less than half the previous price.

Rather than produce yet another flight simulator for the Commodore 64 **SUPERSOFT** have gone one better with the release of **INTERDICTION PILOT**, a space flight simulator. Written by a serving RAF officer, **INTERDICTION PILOT** comes with a comprehensive 48-page manual which provides an insight into the latest technological developments including travel at the speed of light! In simulator mode the trainee pilot can find his way around the controls, and take part in simulated dogfights with alien craft. Many hours of simulator training are necessary before newly commissioned sub-level pilots take part in real life exercises. **INTERDICTION PILOT** is available from **SUPERSOFT** by post, or through dealers at £17.95 (including VAT). A disc version is also available at £19.95.

Why not find out more about Supersoft's range of products by contacting them at Winchester House, Canons Road, Wokingham, Harrow, Middlesex HA9 7PL. Tel: 01-861-7166.

Getting Educated

GET READY TO READ, THE first in a comprehensive series of educational programs, featuring **BJ the Bear**, is now available from Commodore 64. Written by Dr. Richard Eklund, Lecturer in Educational Psychology at the University of Birmingham, and Mrs Lillian Lemons, Headmistress at Moore Mount Nursery and First School in Redditch. Words, the program is available on cassette or disc and is accompanied by a **BJ the Bear** book and a parents' and teachers' manual.

Other programs in the series will include **Get Ready to Think** and **Get Ready for Numbers**. The **Get Ready** program will be followed by **Start to...and Continue to...** for the same subjects but at increasingly advanced levels. The **Get Ready** programs are designed to introduce children of three-to-five years of age to the initial

stages of reading. The **A4** size, full colour "BJ the Bear" book is divided into four levels, each containing a different story about BJ, and is designed to be read to the child in order to prepare him for the learning activities of the program, all of which relate to the Bear's adventures. The book also contains simple pictures to match with letters of the alphabet, a guide to the learning of letters of the alphabet, and join-the-dot and colouring pages. A smiling BJ appears on the screen when the child has made the correct decision, and a frowning BJ appears when the child is incorrect.

The manual, which includes a set of progress charts, instructs the parent or teacher on how to prepare for teaching, how to use the program, how to guide the child through the program, and how to gauge that performance. "Get Ready to Read" is available from Commodore at £19.99.

By Gum!

AFTER **PAQMAN**, NOW welcome **PlaqueMan**, here to tooth invaders, one of Commodore's recent programs for the Commodore 64 home computer. Tooth invaders demands enough computer game-manship to satisfy the most polished young tyro, whilst at the same time offering a worthwhile reminder or denial case — reassurance for Mum and Dad that computer games can have a useful message. As his name suggests, **PlaqueMan**'s mission is to eliminate all the plaque on tooth surfaces. He is guided by joystick and must collect a toothbrush and charge it with fluoride toothpaste to clean the teeth. He must then use dental floss for the gaps in between to complete the cleaning process.

But **PlaqueMan**'s enemy, **D6**, the **Plaque Cavity**, gets ever more cunning and tooth decay sets in

spontaneously, leaving only seconds for remedial treatment. The standard screen shows set of eight teeth — four up, four down — and there is a zoom screen to give a close up view of each tooth as **PlaqueMan** gets to work. **PlaqueMan** has three lives in each game and there are over 50 levels, plus a range of anti-plaque tricks any dentist would be proud of. When all the teeth are completely clean a fluoride rinse cloud obscures and the game moves onto the next still level.

The action is accompanied by music — for triumph or disaster, as appropriate — and there are special sound effects such as brushing noises and "decay alert" warning pips when plaque building becomes critical on untreated teeth. Available on cartridge, **Tooth Invasion** is priced at £59.99.

PILOT'S INSTRUCTION MANUAL



PILOT'S INSTRUCTION MANUAL

INTERDICTION MK III

DATA STATEMENTS

Software Support

HANDIC SOFTWARE ARE obviously firm believers in the potential of the Commodore range of computers used in both the home and in business. Early recently they have produced a range of new programs specifically designed to boost the commercial applications potential of the 64, including such titles as CALC SHEET, DIARY 64 MON 64 (machine code version on cartridge), BILL 64 (for such duties as activating burglar alarms, locks, telephones, central

heating, lighting, etc.) STAT 64 (for statistics and graphics displays), GRAP 64 (for multiplying complicated mathematical functions by other graphs), SUPERBOTS 64 (an expansion board and parallel I/O for the 64), PLOT SWITCH and VIC SWITCH (a multi-user system for the VIC 20 and the 64).

Handic do in fact have a catalogue of all their products, so why not contact them and see what they have to offer? They are based at 5 Albert Road, Crowthorne, Berkshire RG11 3J.

Dealing Yourself

Be successful

Based in Horsham Anagram Systems produce a range of Commodore software aimed at the small to medium sized company. The latest in their range of business accounting packages is Cash Book 64. Based on standard double entry book keeping practice, the system enables the user to maintain accounts to final balance of the month or year to date. The system will also record monthly budget figures and give a budget to actual performance comparison. From a menu of only seven prompts, Cash Book 64 builds a complete accounting record including VAT summary, Audit Trail, Budget Summary and Performance as well as Trial Balance and Cash/Bank book. Cash book 64 costs £75 including VAT.

Going up-market somewhat is the Integrated Accounting System running on Commodore 200 and 400 series machines which has none been enhanced to include Sales Order Processing and full or partial factoring. The 400 is based on standard double entry book keeping practice and comprises Sales, Purchase and General Ledgers with Integrated Stock Control, full Purchase Order Processing and full Sales Order Processing. The full package allows some 200 book keeping operations to be conducted by means of just over 50 screen prompted functions which make it an ideal system for first time users or companies who do not have access to trained accountancy staff. Version 4 with Sales Order Processing and Factoring costs £1595 plus VAT.

Anagram Systems are at 60A Queen Street, Horsham, West Sussex

Back To School

ASK HAVE ADAPTED SEVEN of their highly successful range of mathematics and literacy programs for the BBC Model B Micro to run on the Commodore 64. Available on cartridge at £9.99 is the Autogram 6 Number Patter for the 7-14 year age group. It is an arcade style game with 12 levels of difficulty to test your mental arithmetic. Three cassettes at £11.99 compare Words, Words, Words, it links 30 different words together and the aim is to improve the spelling and vocabulary of children (7-10) who are learning to

read. Number Chaser develops estimation skills through an exciting arcade game. This should be around at Christmas and is £9.99 as a cartridge.

Also available at the moment is Facemaker (building faces according to style) at £4.99, Number Pictorial (puzzles and crosswords with a mathematical bent) at £4.99, Hide and Seek (for improving reading skills and practising team memory retention) at £8.99 and Let's Count (to teach young children to count) at £9.99.

ASK specialise in educational software and the products are marketed by Commodore themselves.

Price War

THEY HAVE ANNOUNCED an entire range of price savings on many of their best selling games for the Commodore 64 and VIC 20. In some cases, Audiogenic has slashed prices by more than half. The price reductions include:

Commodore 64 software: Motor Mania (£8.95 to £5.95), Renaissance (£8.95 to £5.95), Grand Master (Cassette version) (£17.95 to £8.95) and Grandmaster (disc version) (£17.95 to £12.95).

VIC 20 Software: Bones (£7.95 to £5.95) and Grandmaster (on cassette only) (£17.95 to £8.95).

Audiogenic has launched WONTO for the Commodore 64 featuring brilliant full-screen graphics and some of the most devious and vicious monsters ever devised.

WONTO puts the player

in the role of a workman whose job it is to collect bones from different levels of the screen. The various levels are connected by ladders, of which several are unstable giving the player a choice of direction. Unfortunately, the monsters which guard the bones can also climb.

A unique factor of WONTO is in the monster's ability to react to the game with intelligence. The monsters do not just wander about the screen but actually try to trap the player on a particular level.

Once the player has successfully collected all the bones, he progresses to the next level with more valuable bones and more, even meaner monsters.

WONTO is available on cassette at £9.99 from Audiogenic or via the nationwide dealer network.



COMPLNET IS THE NAME of a new service which will allow home computer owners to buy software, use financial and information services and even do everyday shopping direct from their own living room. They will be able to do all this using a simple telephone connection. The key item will be a small box called a Modem, which plugs into the computer and a telephone socket and allows the ordinary home computer to reach and communicate with the COMPLNET service.

Commodore have developed the Modem especially for COMPLNET and an extended 102,000 Commodore 64 owners will have a chance to link up by the end of the year. COMPLNET will offer several key benefits that establish it as a premier service. First, there is no connection time; users need only reach an answer machine or a 11 COMPLNET telephone numbers spread around the country. The modem's auto-dial and software, contained in a built-in memory, will complete the connection in seconds. And once linked to the system, the software directs the user on screen how to log on, or how to log in, or how to log out and display material.



A secondary advantage, and a remarkable development in itself, is the ability to store selected data, transfer of information and programmes, by cutting out the adverse effects of poor telephone lines. Thirdly, up-loading is available. That is to say software and information can be transferred in both directions between the home computer and COMPLNET service. COMPLNET will offer quality software products at discount prices in an area called Software Park. Special features in the Modem and COMPLNET service provided the software being run on any computer other than the one to which it was downloaded. By eliminating the risk of illegal

copying the software protection facility makes the COMPLNET or software service more attractive to suppliers of popular products. Another facility expected to remove computer clubs and hobbyists from an area of the service called 'The Jungle' subscribers can upload software and information they have created themselves directly into the jungle. Other subscribers can then download it on a free or chargeable basis. Seller's earnings less a commission are credited to the COMPLNET account and once a quarter the balance is credited or debited to their bank. Accounts using a Direct Debiting Service. Other facilities include an electronic mailbox service

Commodore product information and alternative software which utilises the Modem to link with remote services.

The Modem costs £19.99 and, as an introductory offer, the first year's subscription to COMPLNET will be free to purchasers. Subscribers have free access to evenings and at weekends and can also use many parts of the service without charge. Later in the year, after an introductory period to familiarise COMPLNET to use and service provider requirements, further mailing and information services will be brought on-stream. These will include catalogue shopping, home financial and insurance services, educational products and information libraries for the domestic environment. COMPLNET has been developed and introduced by Commodore Business Machines (UK) Limited and ADP Networks Services Limited. COMPLNET will be operated on ADP computers and users will be able to link on via local call to the ADP Networks in twelve locations throughout the UK.

Note for more information on modems, log the relevant address (see page 37) or contact us at our Watch Information Centre. You'll find out more of how Commodore's services of modems themselves.

Tandata modems

TANDATA MARKETING have two major modems, the TM100 and TM150 together with Micrograph for the Commodore 64, Commodore PET and LC. The TM 100 operates on 1200 25 baud V21 full duplex mode and is able to store and access data in memory CMOS RAM. It features an eight telephone number menu allowing up to 16 digits per number with a 32-digit reference counter for each number. Transfer of local and/or national or stored numbers will auto recall of unstored numbers. Up to eight separate ID numbers and

gamework each with up to 16 alpha or numeric characters can be permanently and securely stored in the modem's own memory. Also the sometimes lengthy log on procedure is reduced to a single key function.

The TM 120 uses the modem includes all the features of the TM100 and in addition of 1200 25 baud full duplex also offers 12000 baud full duplex and 1200-1200 half duplex to allow two modems to chat. Both modems allow access to Prestel and its 380,000 plus pages of information, electronic mail

and telex facilities, so Telecom Gold, Micronet and Prestel compatible protocols are available. The TM100 costs £165 plus VAT. Software to allow a menu to act as a menu data terminal is now available for the entire range of Commodore micros. For the Commodore 64, software is available on disc, £145 plus VAT, and for ROMs (140 plus VAT) and gives full Prestel emulation with colour display and the inclusion of an on-line message editor. The micrograph includes a program card and comprehensive instruction manual. Soft-

ware for the PET range is available on disc giving full Prestel emulation for the Rescan, Flash, Display, Highlight and Separated Graphics Display modes and on-line message editor is offered. The package comes with a program card and a graphics strip for the VIC 20, the software is on cassette with Prestel emulation for the PET and costs £27 plus VAT.

More information is available from Tandata Marketing, Albion Road, North Mymms, Wotton, Herts. tel: (0844) 68421.

DATA STATEMENTS

New concept

THE CONCEPT KEYBOARD is an original data input device offering many advantages over other keyboard systems. The user programmer has complete flexibility in simply assigning the fixed codes generated from the CONCEPT keyboard matrix to characters, words, shapes, objects or set out on an overlay, thereby giving the user the most efficient keyboard layout for a particular application. The underlying principle is a touch sensitive input selection array-sensitive over the entire matrix area — each touch cell

producing its own unique output code.

Three versions of the CONCEPT keyboard are available:

The A4 unit is 227 x 315 x 25 mm with a 16 x 6 matrix giving 128 touch cells each cell measuring 18 x 18 mm. The two A3 units are 375 x 405 x 25 mm. The A3-128 has a 16 x 8 matrix each cell measuring 24 x 30 mm. With a 16 x 16 matrix the A3-256 has 256 touch cells each 24 x 15 mm.

The codes output from the CONCEPT keyboard are incrementing binary, comprising of only 0 and Head to Head (H) at cell 127 (255) computer with the majority of common code systems — ASCII — ISO



- BINCIC for example. Both positive and negative sockets are available, making the interfacing to any computer system very simple. 1 cell Rollover with 16 cell low standardisation all CONCEPT keyboards. The standard output is 8 bit parallel, but 4 serial output options are available — Parity optically isolated RS232, current loop — RS232C, RS422, RS485. Each is supplied with a switch selectable baud

rate generator, covering the range 50 to 4,500 bauds. The power requirement being very low is usually taken from the host computer... 5 volts at 20mA.

The CONCEPT keyboard is easily interfaced into any computer system and a range of ready made-up interface leads are available for the majority of popular microcomputers including Commodore.

Prices start at £19.00 for the A4 CONCEPT keyboard and £149.00 for the A3-256.

Oric Attack

CREATIVE SPARKS HAS introduced a Commodore 64 version of its highly rated game Oric Attack which is already available for Spectrum and Atari computers.

In Oric Attack, you must defend your castle against the rampaging hordes of Orcs and their lethal crossbows. To defend yourself, you have a broadsword, rocks and your ultimate weapons, set of locking oil. Oric Attack

features four screens of action with excellent sound and original graphics, and is available on cassette for £7.95.

Also available from Creative Sparks for the Commodore 64 are two other games, Black Hawk and Skippy. Black Hawk is a game of action and strategy, featuring the world's deadliest but loneliest aircraft, and Skippy is a greedy, cave dwelling character with the biggest appetite ever.

Creative Sparks games are distributed in the UK through THORN EMI Software Distributors and will be available in all principal retail outlets. Some products will also be available mail order.

Further information is available from THORN EMI Computer Software, Creative Sparks, Thomas House, 296 Lambeth Road, Lambeth, Harris, Tel: 0203 540373.



CERTAINLY A GAME TO have lost respectability to such is the program from New Generation Software called Trashmen. Originally for the Spectrum, it is currently being marketed in the USA for the Commodore 64 under the title of — well for it — Garbage Gobbler. The only change to the name is the game itself is the same as that which was the subject of a recent competition for journalists, won by Computing Today's Editor, Peter Green. He came first playing the game on the Spectrum and won a weekend in Paris, he's just sitting waiting now for the chance to travel transatlantic and throw down the gauntlet to our colonial cousin.

Mole in a hole

Wanted: Manky Mole is a new game from Corelin Graphics for the Commodore 64 featuring a new hero character, Manky Mole. His adversary is the wretched Arthur (who'd have guessed?), safely ensconced in his fortified castle and protected by a bunch of flying jockies. Manky tries to defeat Arthur

by sneaking into the pit and searching coal and in a few effort, steals the secret battle papers and winning tactics. Arthur and his flying jockies have a variety of means at their disposal to prevent Manky from invading. I personally haven't won the game on a screen, and I'm all for innovative games that break away from the usual alien bopping, but the idea seems to me to be in bad taste —

the fact that this game was written by a woman's son only causes me more worry about the current money dispute, not to mention the direction of computer games, regardless of how good they may be technically. The game itself is played at £7.95 and for every game sold, Corelin Graphics will donate 5p to the Manky Welfare Fund. Can we not get out of this political bryl?



Business software can
be expensive and
doesn't always cater to
all your needs. This
series by Grahame
Daxler will give you
an insight into how to
write your own
programs to suit your
requirements.

THIS SERIES WILL ENABLE even the most experienced BASIC programmer to tackle the problem of storing and retrieving data for personal applications. It is based on the CBM 64 connected to a 1041 disc drive but all the techniques and logic used will be applicable to all Commodore machines. Even if you only want to write games, the programming principals and discipline set out in this series will be of great use.

This series will not include a complete listing of a record keeping system but will provide utilities and techniques as well as explanations on how to successfully use disc files (and when to use each type). All the code has been deliberately left in BASIC but the routines may be reentered into machine code if required. I suggest you read the articles and then practice some of the logic and ideas perhaps writing little test programs. Having satisfied yourself that you understand the principle (or at least have got your routines to work) then have a go at writing your own program. Just one other thing to remember — this series is designed for you to be able to write several systems, not just one, so if you find some of the explanations complicated or long-winded do not be put off as you will really get benefits in the end.

DOING IT YOURSELF



Setting out the programs

The first things to note are some of the do's and don'ts of BASIC programming. Firstly, bear in mind when writing that not only have you got to debug your programs, but you may discover a bug in several months time, so make the code readable. Also remember that having got it working, you may want to modify it to store more data and if it is well written, this will be far easier to do. Do not try to put too many commands on one line. Having said that, it is unnecessary to put just one command on a line but with a little thought we can space out some obvious guidelines. Here are a few examples:

```
10 your routine is here
20 rem and here
100 i = 1
200 if i < 5 then 10
```

In this case, we alter the variable 'i' and want to immediately do a test and to know whether to end or not. Having put the test on a new line, we have effectively separated the test. If we come back to this code and add a line 140 for example, although the code may still work the test has become further away from the increment making it more difficult to follow. Here is another:

```
100 for i = 1 to 10
200 if i < 5 then 100
260 your routine is here
300 next
```

In this example we want to go round the loop ten times, but ignoring numbers five and six. The test wants to be on the same line as the loop because there is no reason to have any code between them. Doing this minimises the code size and less prone to bugs if something is added.

Putting so many statements on one line saves memory and marginally speeds up the code but these advantages are insignificant and are far

outweighed by the disadvantages — do not try to be memory efficient in this way.

We will now examine the REM command. This command is an essential programming tool but must be used sensibly:

```
100 m = 0: rem number of
to zero
```

This is hopeless! The REM isn't exactly the same as the code. Much better would be:

```
100 m = 0: rem number of
lines
```

This tells us what 'm' is being used for. The code itself tells us that it is being set to zero, so why put it in a REM? Experience will show when

now let's decide what variable names to use. I advise having temporary variables which can be used to pass data to a subroutine and to be returned back from subroutines. Also for convenience FOR...NEXT loops, try to use the same variables and avoid using any variable that you use in a subroutine. This is a common problem which causes FOR...NEXT loops to end suddenly or never end! For all the new flaggers, that is to say all new variables that are specific to the program being written and not in common subroutines, use two letters (or more) because this is more descriptive and easier to follow. 'AM' for amount is absolutely quite clear but

then amend your program so that it is neat and easy to follow. In this way the benefits will become more obvious.

Getting framed

If your first program is successful and you want to write more, you will need a program frame containing your favourite tried and tested routines. I will call this program a FRT. In your next you will need subroutines for screen handling, disk handling, error messages and all the little things that you only want to write once. After a while you may add to them or improve them making them more elaborate. By keeping these in a net you



to use REMs and what to put in them. For now, just remember that as you write a lot of code you understand it fully but in two months' time it will be harder to follow, so use some REMs telling you what variables are expected for the routine and what variables are returned. REMs do use memory and marginally slow the program — ignore these facts and use as many as you like. Try to format them neatly on the screen — leave spaces if necessary so that they may be easily read.

```
100 am = 0: rem          exchange rate
200 m = 0: rem          number of things
260 for i = 1 to 100: rem wait loop
```

'FF' for amount is obscure (long more than two letters makes the code even clearer 'FAMOUNT' for amount) but is unnecessary and has some disadvantages — it uses memory, it leads to more syntax errors as BASIC may discover a reserved word such as 'ON' in 'MCINDAYS' and in many cases BASIC only stores the last two characters of the variable name so the variables 'FAMOUNT' and 'FAMIND' are the same.

When you have completed your program, go back and read the above.

We will have easy access to them and the subroutines will be in the same place in every program making each program easier to understand. As your routines get more advanced you will find them becoming like a set of advanced BASIC commands and you will really be setting up a tailor-made high level language (well higher than BASIC anyway).

The line numbers you are a matter of personal preference but there are still some things to remember: do not start at line zero or one or ten etc — you can have line numbers up to 65535 so make good use of them. Any routines listed in this series will have line numbers between 1000 and 3999 but you may relocate them (changing all the relevant jumps etc) where you wish.



Commodore have
launched a new
business machine with
a full 128K of RAM
and some interesting
features. Will it appeal
to businessmen?
Simon Dismore
investigates.

IS BIGGER BETTER?

COMMODORE'S TOP-OF-the-range machine, the 8296D, is an interesting mixture of the old and the new. Underneath the "boxed iconom" casing is a system which is fully compatible with the earlier products in the 8000 range, but carries a full 128K of RAM, organised in blocks of 16K.

"Very nice!" we thought, as our evaluation machine was unloaded from the container lorry (Commodore have not skimped on protective packaging). Our first question was "what can we do with all this memory?"—and here the answers were a little disappointing.

You might wish to use a computer either for developing your own

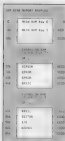


Memory blocks

Figure 1a

Bit 7:	ENABLE EXPANSION MEMORY 0 disabled (default on power-up) 1 enabled	Bit 2:	SELECT MEMORY Addresses 8000 - 8FFFF 0 block 0 1 block 1
Bit 6:	I/O PEAK THROUGH Addresses 8000 - 8FFFF 0 disabled 1 enabled	Bit 1:	WRITE PROTECT EXPANSION RAM Addresses 8000 - 8FFFF 0 not protected 1 protected (I/O is not protected if peak through is enabled)
Bit 5:	SCREEN PEAK THROUGH 0 disabled 1 enabled	Bit 0:	WRITE PROTECT EXPANSION RAM Addresses 8000 - 8FFFF 0 not protected 1 protected (screen is not protected if peak through is enabled)
Bit 3:	SELECT MEMORY Addresses 8000 - 8FFFF 0 block 0 1 block 1	COM reserve bit 4 for future applications	





10

programs, as the running costs of these programs are 90% of purchases fall into the latter category, and we were surprised to find that there was no software available that took advantage of the extra memory. One might expect that products might be in short supply for what is a relatively new machine—but the complete absence of appropriate programs was quite baffling. What can your friendly Commodore dealer demonstrate except the existing range of software, none of which was written with the SBC in

front in the pipeline, and Cammenga would be Hamid Schwarz's CalcRoute spreadsheet program. CalcRoute for the 8286 is still a few weeks away from completion, but the authors very kindly offered to share their experiences with us, even to the extent of getting their programming manager to call us from Sweden.

They left out the extra memory was "useful but not essential" for their test program. The benefit is that larger spreadsheets with more data can be manipulated in the memory of the system, while maintaining compatibility with data on earlier machines.

Once Caltrans was up and running, their word processing package might follow. An interesting possibility would be the use of the expensive RAI as a "placeholder" to hold results while programs were being changed — the first step towards an integrated environment of the sort offered by AppleWorks and Lotus 3-1-1.

Harrier also pointed out that they had to go onto the machine with a soldering iron to strap the appropriate internal jumpers so that they could select between expansion memory and firmware under software control (jane strapped, a USB rail can be used to enable and disable any available EPROMs or ROMs). Your dealer could probably do this for you, however. We would be interested to hear any other experiences in this area—perhaps Commodore should add new chips which can be used to configure

Because the expansion memory is configured in fixed 1MB blocks, you would have to erase your own memory allocation routines, before attempting to access the memory for any sophisticated approach (the clipboard idea would certainly not use the

memory as an ultra-fast disk drive. If you are converting a word processing or spreadsheet package, you will probably have existing routines which spend most of their time with little modification. Otherwise, you should be careful to design your allocation routines (which must be permanently in the lower 128 megabytes) as early stages of development. One final word of advice from Hander—interruptions are truly a program, jumpstart, and interrupting aside, writing memory banks is easy.

performed by FOX using a selection using the intensity control register at 5400 (see Figure 1).

The 5296 also gives you four "pages" of video memory, instead of the usual one. This could be very useful, and is much easier to program. Commodore suggests that you might use the additional screens for help screens and menus, though once again, shipping might be required if you have filled extra

The alternate screen, just selected through a menu



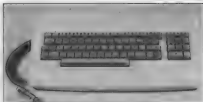
Abstract

ling `FOCUS` sequence. Unfortunately, the `FOCUS` statement in the resident basic (4.0) appears to work with only the default screen. Commodore number the screens from 1 to 4 in their documentation, but you will probably prefer to use a range from 0 to 3 for ease of programming — in either case, the default screen is the lower number). This presumably means that you have to use the `FOCUS` statement to get characters onto any of the alternate screens. Labels are passed something in Commodore's file, then wonderful, in usual documentation, the screen to be a major windowing block to production use of the system screen.

So that is as easy, as it appears, but the data level that the real-time processors were working on was a small amount of RAM, presented to them by the Base, represented some complex constraints, added a variation for users handling it. Figure 3 shows, the PDCI sequence for selecting between the four sources, where figures 1 and 4 illustrate a full screen copy and a partial copy, respectively. We learned too, on our own that the 8255 does NOT take kindly to errors in PDCI state changes, so we have shown some ways to deal with errors, and we demonstrated and trapped (see Figure 5).

The most exciting applications for multiple processors is almost certainly "workstation." You run it thus with our parallel server program. The 6386 client screens 8 on power-up but leaves the other screens with random colors, so you won't probably wish to run them on black as my looping FORT 80 memory file allocate screen is NOT cleared when you press reset, so any data you put in there is safe from everything except a power

The resident BASIC is very slow (over 10 seconds to copy a somewhat complex set of all remarks removed and multiple statements on a line), so you would almost certainly want to re-code any worthwhile routine.



into machine code once you have them working properly. Other amusing experiments include reversing rows and columns and simple text editing using the alternate screen in buffer.

If you are already a Compaqware user and have a set of 8000 disk drives, you might find the 8296 an interesting machine. It certainly doesn't seem

North replacing the 8008 with an 8086 just to get the benefits of additional memory for which no software is yet written.

If you don't already have a machine, would you consider the \$295-DiWeed? Corrosion has produced a repeatable but not a perfect product, which couldn't be regarded as "state of the art" in any respect.

January 2010 computer sales were approximately 5000, on each side of the river, a figure which compares well with most other U.S. cities. Washington life was under DOW 40 in 1986, and there is room for up to 234 of sector seven, both of which should be satisfactory for most purposes.

CPU: The 486 remains cannot compare with 486-based Value products.

References

ROUTINE TO COPY SPECIAL DISK FILES TO APPROPRIATE
CLUSTERS FOR "LOADING" SOFTWARE

```

1573 rem Subroutine to copy a partial row of data
1574 rem On entry, variables passed source 's' and destination 'd' address
1575 rem variables 's1', 's2', 's3' give source coordinates
1576 rem variables 'd1', 'd2' give destination starting point
1577 rem Variables 's1' and 's2' are in lines 1102,1130
1578 rem Swap 's1' and 's2' to always top left, bottom right...
1579 rem If played then always slow
1580 rem If played then playing slow
1581 rem Subroutine 's1' and...
1582 rem If 's1' is Col=50 or Col=60 or Col=70 then s1='1,4,8,9'; goto 8080
1583 rem If 's1' is Col=50 or Col=60 or Col=70 then s1='2,5'; goto 8080
1584 rem Subroutine of 's1' 's2' to produce an illegal move
1585 rem If 's1' is Col=50 or Col=60 or Col=70 then s2='1,4,8,9'; goto 8080
1586 rem If 's1' is Col=50 or Col=60 or Col=70 then s2='2,5'; goto 8080
1587 rem Subroutine parameters 's' and 'd' defined as... goto 1102,1130
1588 rem s1=s1+1; goto 8080; goto 8080; rem s1=s1+1; goto 8080; goto 8080
1589 rem s1=s1+1; goto 8080; goto 8080; rem s1=s1+1; goto 8080; goto 8080
1590 rem Now copy line for time
1591 for j=0 to Col-1
1592   for i=0 to Col-1
1593     copyDest(i,j)=copyDest(i,j)
1594   next j
1595 next i
1596 rem loop completed
1597 return

```



like the 6800 series, it runs slowly and has a limited addressing range. You might start to worry about limitations after a while — it depends whether you expect that your applications will grow substantially within the lifetime of your purchase.

Screen: Contrast and character sets are both good, and you can tilt and scroll the screen as required. However, we did find that the pronounced curvature of the screen and the high persistence phosphor took a bit of getting used to. This is simply a matter of individual preference, rather than a serious design flaw.

The review machine needed to where like an old man-of-war TV — something which your store would presumably adjust before delivering the system.

Keyboard: The keyboard has a very 'springy' feel which would probably irritate touch typists at first. Keys must be depressed fully before a character is sent, so there is little risk of mistyping. We did feel that Commodore should have provided a full four-key cursor pad, rather than using shifts to generate cursor left and up/down/other opposites, given that this product is obviously intended for the serious business market.

We also thought that there should be an Enter key on the numeric keypad (generating a carriage return character). There is certainly room for these features on the detachable keyboard unit, which is one of the larger on the market.

As we go to press, Commodore are preparing plans for promoting the 6264-D from September

onwards. The price of the system will be £1899 including VAT for the standard two-drive configuration. This price will include three 'bundled' software packages covering the main applications areas. These consist of SuperScript, a word processor and spelling checker; The Manager, a database filing package; and Calcitruit, a spreadsheet program. Bought separately, these would cost about £100. This makes the 6264-D reasonable value for money at the moment — though we are a little concerned that it could lose ground to some of the more up-to-date 16-bit products which will appear over the next six to nine months.

Hardik's upgraded Calcitruit will also appear in September, in a version which is compatible with all 6800 series machines.

For further details consult your extremely helpful team on 01844 778800.

As to availability, Commodore tell us that 250 dealers throughout the country will be stocking the 6264. No new peripherals are being launched for the machine, but three compatible printers are available. Two are tractor feed dot matrix types with 40 cps/80 columns and 750 cps/132 columns respectively, while the third is a daisy-wheel tractor feed printer working at 40 cps/136 columns. If you want the daisy-wheel version it will cost you £600 including VAT.

Finally, a message to Software House who are planning to use the expansion memory on the pagged screen. Please drop us a line and tell us what you are up to, so that we can follow the progress of the 6264 in future issues.

Error trapping

COMBINE ERROR TRAP ROUTINE TO DISPLAY MESSAGE ON PAGE 0

```

1000 rem Error trap with message in variable 'a$'
1010 rem Display diagnostic
1020 print "Line 11.1844'junk' was"
1030 rem Set last page 0 for clearing
1040 spcs=spcs-1000
1050 goto 1000

```

Our fearless reviewer,
Dave Crisp, gives you
his general view of
Superbase for the
Commodore 64.

SUPERBASE OVERVIEW

ON MY COMMODORE 8096 I use the SILICON OFFICE and on my 5444 I use SUPERBASE. If you have never used programs like the above before, here is a brief description of what they are used for.

In their basic form they are best described as a pre-written database program; their power however comes when you want to cross reference information from one file to another, input new information and to generate a REPORT compiled from information from one or more files. Imagine for example three drawers of a filing cabinet. Each drawer contains a set of files.

eg. Drawer 1 - Names and addresses

Drawer 2 - Personal data (salary, car etc.)

Drawer 3 - Work record

If the company wanted to send a standard letter to all employees who satisfy a particular set of conditions it would normally be a long and difficult job checking and cross checking information from the three drawers. With a program of the above type however a small sub-program could be written to examine all files, obtain and create a list of all the people who meet the requirements. The list could then be used to personalise the stock letter which would have been created on a word processor. That is one use of a program like the above. Other uses are mailing lists, patient records, invoicing etc.

A program which is used to fulfil a specialised function through SUPERBASE and programs like it is called an APPLICATION

and it is often possible to buy an applications package to run with the database management system.

Superbase

I will not try and describe how to use Superbase as it is a very comprehensive program. I will just give an overview of it and a summary of how I have been able to use it. I had already learned to use the SILICON OFFICE and so when I first came to use SUPERBASE I was pleased to see that there were many similarities. Within a couple of weeks I had produced a stock control application linked to an electronic till drawer which could cope with about 700 stock items and produce daily sales lists, low stock items, price lists, etc, etc. SUPERBASE can be used HARD, by this I mean that it can be used as a menu driven database with all commands entered direct. By using the menu commands you soon get the idea of manipulating data in files and it is then a simple job of linking these direct commands together to form a program.

With discs

In use with a single disc drive it is possible to have up to 15 separate data files and also any programs that you may have produced. The number of records in each file is, in theory unlimited but it is of course restricted by the 1MB restriction of the 5444 drive. It is possible and also easy to use two drives and using this system I have not yet found any limitations. With two



drives, it is easy to make backup copies of your data. Initially I found I was not bothering to take copies of my data but the dreaded day came when with the help of a split cup of coffee about three weeks' worth of data had gone. One of the interesting aspects of SUPERBASE is that it will link up to EASYSOFT and so personalised letters and documents are reasonably easy to create. I say

reasonably easy to create because this is not a program that you would be able to just load and run — you would need to spend quite a few hours learning to look and manipulate files and to program effectively.

Basically speaking...

The language used by SUPERBASE is a mixture of BASIC plus some of its own commands. Its own

commands can be entered in an abbreviated form or in a sentence, eg. to create a list of all records from a particular file you can write into a program: display all records from file a name/ and to create a specialised list you can simply say: find list where town = town name/.

This is the easy way to input commands but it is effective and once you have the idea you can input commands in an abbreviated form. The abbreviations are logical and are still meaningful when you go back to a program some time later.

There are built in pages called help screens which you can call up and these pages give you the syntax and use of most of the important commands available. On the whole it is user friendly program which is only limited in use by your imagination. Some of the important features are listed below with a brief explanation of their function.

1. BATCH. This allows the whole file to be updated automatically for instance if the VAT rate changed it would be possible to update each record with a simple list of commands.

2. FILE SIZE. It is possible to increase the length of a record without losing any data. This is a very useful feature and one which should be available on any good database.

3. FIND LIST. This allows you to create a specialised list of things, which usually a given set of conditions. There can be as many conditions as you want and the created list can be used and destroyed or kept for future use.

4. DATE. This function is very useful. It enables you to set up conditions that must be met by a given date. Very useful on invoice/statement type applications.

It is possible if you have written an application with SUPERBASE to PROTECT it so that the program cannot be hired and also to pro-

word parts of the program to restrict unauthorised use, a feature which I have found useful to restrict access of personal details while allowing basic name address type information to be readily accessible. Screen colours can be changed easily and sound can be used to draw attention to expected inputs. There is a built in software printer interface and so you should be able to use most of the common printers and switching from printer to screen is simply made if when writing a program you want to see the effects of a report on screen rather than on the printer. It is just a matter of replacing the word PRINT with DISPLAY in the program. This can save pumping out reams of paper during debugging.

Great stuff

This is a program that I would not like to be without. It is versatile and probably the most useful of all my software. It costs about £100.00 but is considerably cheaper than its big brothers on larger machines. One thing that is important is when I had a problem with one of the programs, PRECISION software went very keen and quick to help. They do seem concerned with after sales service, something which is sadly lacking in many soft/hardware companies.

Doesn't amongst you may be interested to know that there is a demonstration of it available which shows much of the potential of SUPERBASE.

If you are in business or run a club/society have large collections or an unruly filing cabinet or just enjoy databases then this is the one.

64⁺



Doesn't your 64 deserve a good book?

To make full use of the your Commodore 64's powerful capabilities, you need the help of good books. You need SAMS books - they'll keep you and your micro busy for hours.

Commodore 64 Programmer's Reference Guide

A bestselling book packed with professional tips to help you program your 64 successfully and creatively. Now at a new low price.

400 pages/ISBN 0 672 22056-5/£19.95

Commodore 64 Graphics and Sounds

Timothy Ott Knight

A combination of book and software to help you master the 64's powerful graphics and sound capabilities and use them in spectacular routines.

Book: 112 pages/ISBN 0 672 22170 7/£10.95

Book with software: ISBN 0 672 26780 3
£19.95 + VAT

Commodore 64 BASIC Programs

Knight and Lobell

A well-illustrated collection of fun and practice programs, each of which is explained in full so that you learn how it works as you use it.

Book: 176 pages/ISBN 0 672 22171 3/£12.95

Book with software: ISBN 0 672 26777 5/
£13.50 + VAT

Learn BASIC Programming in 14 days on your Commodore 64

Ol Schreiner

An easy-to-use and rapid method of learning how to program your 64, consisting of 14 chapters each of which can be covered in a day.

192 pages/ISBN 0 672 22079-6/£19.95

Available from all good bookshops.

SAMS books are distributed in the UK by Pitman Publishing, 108 Long Acres, London EC2B 5AA (Tel: 01-776 3340).

Pitman

Superbase 96

Superbase 96 is a powerful database management system for the Commodore 64. It allows you to create, store, and retrieve data in a structured and efficient manner. The system is designed to be easy to use and to integrate with other software applications.

Superscript II

Superscript II is a word processing software package for the Commodore 64. It offers a range of features including text formatting, spell checking, and file management. The interface is intuitive and designed for ease of use.

Master

Master is a comprehensive software suite for the Commodore 64. It includes modules for accounting, inventory management, and customer relationship management. The system is designed to streamline business operations.

*** Start of January ***	*** 31st Dec 1991 ***	*** 31st Dec 1992 ***
Master 96 (Book)	£19.95	£19.95
Master 96 (Software)	£19.95	£19.95
Master 96 (Book & Software)	£39.90	£39.90
Superscript II (Book)	£12.95	£12.95
Superscript II (Software)	£12.95	£12.95
Superscript II (Book & Software)	£25.90	£25.90



Unit 120, 100, Kingston Hill, Surrey, KT1 1ST, UK. Tel: 01-847-7054

LIGHT TO THE POINT



SPECTRUM AND COMMODORE 64

£17.25

DRAGON TANDY

£11.50

Inclusive
NO-INTERFACE REQUIRED

Discover the exciting world of creating your own graphics on screen.

The Trojan Light Pen will draw boxes, circles, lines, freehand pictures, curves and boxed pictures with full screen facility.

All in-Hi-Res screen in any of 4 colours for the Dragon/Tandy, 8 colours for the Spectrum, and 16 colours for the Commodore 64.

For educational or business use

CALL 01-847-7054

TROJAN

Micro Computer Software & Accessories

Send us your order to
TROJAN PRODUCTS

105, Dethyde, Quainton, Bucks MK22 7TJ
Tel: (01295) 205491

Yours

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

Whatever your interest in the Commodore range of computers and peripherals, you simply cannot afford to miss a single issue. The magazine is packed with news, reviews of the latest

And it's so easy to do! Just fill out the form below, write a cheque and send it off to the Subscriptions department. You know it makes sense!

[illegible]

**** COMMODORE 64 ****

R.S.B. (Rapid Backup System)

CONVERT FOUR SINGLE AND MULTITASK TAP
PROGRAMS TO TURBO-LOAD FASTER THAN THE
COMMODORE DISK DRIVE. EACH CONVERTED PROGRAM
WILL LOAD INDEPENDENTLY OF R.S.B. NO FORKS, NO
SAY CALLS, NO USER KNOWLEDGE REQUIRED. LOAD
"THE HORNET" IN 10 SECS. FOR PERSONAL USE ONLY.
CASSETTE CT 20 INC.

HYPERSAVE - 84

A VERSATILE AND FLEXIBLE FAST-LOADING UTILITY &
ADDITIONAL BASIC COMMANDS. SAVE/LOAD/VERIFY
UP TO 48K CONTIGUOUS RAM AT HYPERFREQ.
CASSETTE CT 20 INC.

BACK-UP - DELUXE

A POWERFUL AND USER FRIENDLY SECURITY BACKUP
UTILITY. LOAD/SAVE ALL PROGRAM PARTS
CONTINUOUSLY. INCORPORATES A COMPLICATED
HEADER READER. FOR PERSONAL USE ONLY.
CASSETTE CT 20 INC.

PRG-SPRITS

SPRITE LIBRARY. DESIGN, MANIPULATE, CATALOGUE
MULTI AND SINGLE COLOR SPRITES. CREATE DATA,
JOYSTICK CONTROL. PACKED WITH FEATURES! 100%
MACHINE CODE. CASSETTE CT 20 INC. COMPLIES WITH
YOUR FIRST SPRITE LIBRARY.

SPECIAL DISCOUNT: ANY TWO PROGRAMS £3.99

ALL FOUR £7.99 PLUS POST! SECRETS!

100 QUESTIONS WITH EVERY ORDER.

CHECKOUT P.O. OR SAE 500.

DOSOFT

2 OAKMOOR AVENUE
BLACKPOOL FY2 0EE

*** FAST DISPATCH GUARANTEED ****

BUNTASOFT

GREAT ADVENTURES. STUPIDLY LOW PRICES!
BLACK KINGDOM - MEDIEVAL TEXT
ADVENTURE FOR COMMODORE 64 AND MSX
VIC 20 (IN 2 1/2K EPISODES!)

***** FIVE STAR QUALITY *****

HOME COMPUTING WEEKLY 24 99

SEA MOONED - SPACEAGE TEXT

ADVENTURE NO. SOUND EFFECTS

164 VIC 20

ONLY £4.50

"GOOD VALUE..." - RINGO ADVENTURES

S.A.E. FOR COMPLETE LIST

1 CRIPPING HALL COTTES

CRIPPING HEATHS S69 0PS



If an advertisement
is wrong we're here
to put it right.

If you see an advertisement in the press, in print,
on posters or in the cinema which you find
unacceptable, write to us at the address below.

The Advertising Standards Authority



ASA Ltd, One York House, Torrington Place, London WC1E 7HN

Supersoft 64

WE HAVE THE CHOICE OF 100+ 4 KILL
PROGRAMS INCLUDING: BATTLESHIP, CHUCKY,
TIGER, AND 100+ MORE! (SEE LISTING FOR FULL
LIST OF 100+ TITLES) ... SEE US AT THE
FUTURE FEST EXHIBITION, 10/11/84, 10/12/84,
10/13/84, 10/14/84, 10/15/84, 10/16/84, 10/17/84,
10/18/84, 10/19/84, 10/20/84, 10/21/84, 10/22/84,
10/23/84, 10/24/84, 10/25/84, 10/26/84, 10/27/84,
10/28/84, 10/29/84, 10/30/84, 10/31/84, 11/1/84,
11/2/84, 11/3/84, 11/4/84, 11/5/84, 11/6/84, 11/7/84,
11/8/84, 11/9/84, 11/10/84, 11/11/84, 11/12/84,
11/13/84, 11/14/84, 11/15/84, 11/16/84, 11/17/84,
11/18/84, 11/19/84, 11/20/84, 11/21/84, 11/22/84,
11/23/84, 11/24/84, 11/25/84, 11/26/84, 11/27/84,
11/28/84, 11/29/84, 11/30/84, 12/1/84, 12/2/84,
12/3/84, 12/4/84, 12/5/84, 12/6/84, 12/7/84, 12/8/84,
12/9/84, 12/10/84, 12/11/84, 12/12/84, 12/13/84,
12/14/84, 12/15/84, 12/16/84, 12/17/84, 12/18/84,
12/19/84, 12/20/84, 12/21/84, 12/22/84, 12/23/84,
12/24/84, 12/25/84, 12/26/84, 12/27/84, 12/28/84,
12/29/84, 12/30/84, 12/31/84.

Figaro

FIGARO

WE HAVE THE CHOICE OF 100+ 4 KILL
PROGRAMS INCLUDING: BATTLESHIP, CHUCKY,
TIGER, AND 100+ MORE! (SEE LISTING FOR FULL
LIST OF 100+ TITLES) ... SEE US AT THE
FUTURE FEST EXHIBITION, 10/11/84, 10/12/84,
10/13/84, 10/14/84, 10/15/84, 10/16/84, 10/17/84,
10/18/84, 10/19/84, 10/20/84, 10/21/84, 10/22/84,
10/23/84, 10/24/84, 10/25/84, 10/26/84, 10/27/84,
10/28/84, 10/29/84, 10/30/84, 10/31/84, 11/1/84,
11/2/84, 11/3/84, 11/4/84, 11/5/84, 11/6/84, 11/7/84,
11/8/84, 11/9/84, 11/10/84, 11/11/84, 11/12/84,
11/13/84, 11/14/84, 11/15/84, 11/16/84, 11/17/84,
11/18/84, 11/19/84, 11/20/84, 11/21/84, 11/22/84,
11/23/84, 11/24/84, 11/25/84, 11/26/84, 11/27/84,
11/28/84, 11/29/84, 11/30/84, 12/1/84, 12/2/84,
12/3/84, 12/4/84, 12/5/84, 12/6/84, 12/7/84, 12/8/84,
12/9/84, 12/10/84, 12/11/84, 12/12/84, 12/13/84,
12/14/84, 12/15/84, 12/16/84, 12/17/84, 12/18/84,
12/19/84, 12/20/84, 12/21/84, 12/22/84, 12/23/84,
12/24/84, 12/25/84, 12/26/84, 12/27/84, 12/28/84,
12/29/84, 12/30/84, 12/31/84.

WE HAVE THE CHOICE OF 100+ 4 KILL
PROGRAMS INCLUDING: BATTLESHIP, CHUCKY,
TIGER, AND 100+ MORE! (SEE LISTING FOR FULL
LIST OF 100+ TITLES) ... SEE US AT THE
FUTURE FEST EXHIBITION, 10/11/84, 10/12/84,
10/13/84, 10/14/84, 10/15/84, 10/16/84, 10/17/84,
10/18/84, 10/19/84, 10/20/84, 10/21/84, 10/22/84,
10/23/84, 10/24/84, 10/25/84, 10/26/84, 10/27/84,
10/28/84, 10/29/84, 10/30/84, 10/31/84, 11/1/84,
11/2/84, 11/3/84, 11/4/84, 11/5/84, 11/6/84, 11/7/84,
11/8/84, 11/9/84, 11/10/84, 11/11/84, 11/12/84,
11/13/84, 11/14/84, 11/15/84, 11/16/84, 11/17/84,
11/18/84, 11/19/84, 11/20/84, 11/21/84, 11/22/84,
11/23/84, 11/24/84, 11/25/84, 11/26/84, 11/27/84,
11/28/84, 11/29/84, 11/30/84, 12/1/84, 12/2/84,
12/3/84, 12/4/84, 12/5/84, 12/6/84, 12/7/84, 12/8/84,
12/9/84, 12/10/84, 12/11/84, 12/12/84, 12/13/84,
12/14/84, 12/15/84, 12/16/84, 12/17/84, 12/18/84,
12/19/84, 12/20/84, 12/21/84, 12/22/84, 12/23/84,
12/24/84, 12/25/84, 12/26/84, 12/27/84, 12/28/84,
12/29/84, 12/30/84, 12/31/84.

Vizawite 64

VIZAWITE 64

WE HAVE THE CHOICE OF 100+ 4 KILL
PROGRAMS INCLUDING: BATTLESHIP, CHUCKY,
TIGER, AND 100+ MORE! (SEE LISTING FOR FULL
LIST OF 100+ TITLES) ... SEE US AT THE
FUTURE FEST EXHIBITION, 10/11/84, 10/12/84,
10/13/84, 10/14/84, 10/15/84, 10/16/84, 10/17/84,
10/18/84, 10/19/84, 10/20/84, 10/21/84, 10/22/84,
10/23/84, 10/24/84, 10/25/84, 10/26/84, 10/27/84,
10/28/84, 10/29/84, 10/30/84, 10/31/84, 11/1/84,
11/2/84, 11/3/84, 11/4/84, 11/5/84, 11/6/84, 11/7/84,
11/8/84, 11/9/84, 11/10/84, 11/11/84, 11/12/84,
11/13/84, 11/14/84, 11/15/84, 11/16/84, 11/17/84,
11/18/84, 11/19/84, 11/20/84, 11/21/84, 11/22/84,
11/23/84, 11/24/84, 11/25/84, 11/26/84, 11/27/84,
11/28/84, 11/29/84, 11/30/84, 12/1/84, 12/2/84,
12/3/84, 12/4/84, 12/5/84, 12/6/84, 12/7/84, 12/8/84,
12/9/84, 12/10/84, 12/11/84, 12/12/84, 12/13/84,
12/14/84, 12/15/84, 12/16/84, 12/17/84, 12/18/84,
12/19/84, 12/20/84, 12/21/84, 12/22/84, 12/23/84,
12/24/84, 12/25/84, 12/26/84, 12/27/84, 12/28/84,
12/29/84, 12/30/84, 12/31/84.

WE HAVE THE CHOICE OF 100+ 4 KILL
PROGRAMS INCLUDING: BATTLESHIP, CHUCKY,
TIGER, AND 100+ MORE! (SEE LISTING FOR FULL
LIST OF 100+ TITLES) ... SEE US AT THE
FUTURE FEST EXHIBITION, 10/11/84, 10/12/84,
10/13/84, 10/14/84, 10/15/84, 10/16/84, 10/17/84,
10/18/84, 10/19/84, 10/20/84, 10/21/84, 10/22/84,
10/23/84, 10/24/84, 10/25/84, 10/26/84, 10/27/84,
10/28/84, 10/29/84, 10/30/84, 10/31/84, 11/1/84,
11/2/84, 11/3/84, 11/4/84, 11/5/84, 11/6/84, 11/7/84,
11/8/84, 11/9/84, 11/10/84, 11/11/84, 11/12/84,
11/13/84, 11/14/84, 11/15/84, 11/16/84, 11/17/84,
11/18/84, 11/19/84, 11/20/84, 11/21/84, 11/22/84,
11/23/84, 11/24/84, 11/25/84, 11/26/84, 11/27/84,
11/28/84, 11/29/84, 11/30/84, 12/1/84, 12/2/84,
12/3/84, 12/4/84, 12/5/84, 12/6/84, 12/7/84, 12/8/84,
12/9/84, 12/10/84, 12/11/84, 12/12/84, 12/13/84,
12/14/84, 12/15/84, 12/16/84, 12/17/84, 12/18/84,
12/19/84, 12/20/84, 12/21/84, 12/22/84, 12/23/84,
12/24/84, 12/25/84, 12/26/84, 12/27/84, 12/28/84,
12/29/84, 12/30/84, 12/31/84.

*** 100% 100% 100% *** 100% 100% 100% *** 100% 100% 100% ***

PROGRAM	PRICE	DATE	DATE	DATE	DATE	DATE	DATE	DATE	DATE
BATTLESHIP	£1.99	10/11/84	10/12/84	10/13/84	10/14/84	10/15/84	10/16/84	10/17/84	10/18/84
CHUCKY	£1.99	10/11/84	10/12/84	10/13/84	10/14/84	10/15/84	10/16/84	10/17/84	10/18/84
TIGER	£1.99	10/11/84	10/12/84	10/13/84	10/14/84	10/15/84	10/16/84	10/17/84	10/18/84
100+ MORE	£1.99	10/11/84	10/12/84	10/13/84	10/14/84	10/15/84	10/16/84	10/17/84	10/18/84

WE HAVE THE CHOICE OF 100+ 4 KILL
PROGRAMS INCLUDING: BATTLESHIP, CHUCKY,
TIGER, AND 100+ MORE! (SEE LISTING FOR FULL
LIST OF 100+ TITLES) ... SEE US AT THE
FUTURE FEST EXHIBITION, 10/11/84, 10/12/84,
10/13/84, 10/14/84, 10/15/84, 10/16/84, 10/17/84,
10/18/84, 10/19/84, 10/20/84, 10/21/84, 10/22/84,
10/23/84, 10/24/84, 10/25/84, 10/26/84, 10/27/84,
10/28/84, 10/29/84, 10/30/84, 10/31/84, 11/1/84,
11/2/84, 11/3/84, 11/4/84, 11/5/84, 11/6/84, 11/7/84,
11/8/84, 11/9/84, 11/10/84, 11/11/84, 11/12/84,
11/13/84, 11/14/84, 11/15/84, 11/16/84, 11/17/84,
11/18/84, 11/19/84, 11/20/84, 11/21/84, 11/22/84,
11/23/84, 11/24/84, 11/25/84, 11/26/84, 11/27/84,
11/28/84, 11/29/84, 11/30/84, 12/1/84, 12/2/84,
12/3/84, 12/4/84, 12/5/84, 12/6/84, 12/7/84, 12/8/84,
12/9/84, 12/10/84, 12/11/84, 12/12/84, 12/13/84,
12/14/84, 12/15/84, 12/16/84, 12/17/84, 12/18/84,
12/19/84, 12/20/84, 12/21/84, 12/22/84, 12/23/84,
12/24/84, 12/25/84, 12/26/84, 12/27/84, 12/28/84,
12/29/84, 12/30/84, 12/31/84.



WE HAVE THE CHOICE OF 100+ 4 KILL
PROGRAMS INCLUDING: BATTLESHIP, CHUCKY,
TIGER, AND 100+ MORE! (SEE LISTING FOR FULL
LIST OF 100+ TITLES) ... SEE US AT THE
FUTURE FEST EXHIBITION, 10/11/84, 10/12/84,
10/13/84, 10/14/84, 10/15/84, 10/16/84, 10/17/84,
10/18/84, 10/19/84, 10/20/84, 10/21/84, 10/22/84,
10/23/84, 10/24/84, 10/25/84, 10/26/84, 10/27/84,
10/28/84, 10/29/84, 10/30/84, 10/31/84, 11/1/84,
11/2/84, 11/3/84, 11/4/84, 11/5/84, 11/6/84, 11/7/84,
11/8/84, 11/9/84, 11/10/84, 11/11/84, 11/12/84,
11/13/84, 11/14/84, 11/15/84, 11/16/84, 11/17/84,
11/18/84, 11/19/84, 11/20/84, 11/21/84, 11/22/84,
11/23/84, 11/24/84, 11/25/84, 11/26/84, 11/27/84,
11/28/84, 11/29/84, 11/30/84, 12/1/84, 12/2/84,
12/3/84, 12/4/84, 12/5/84, 12/6/84, 12/7/84, 12/8/84,
12/9/84, 12/10/84, 12/11/84, 12/12/84, 12/13/84,
12/14/84, 12/15/84, 12/16/84, 12/17/84, 12/18/84,
12/19/84, 12/20/84, 12/21/84, 12/22/84, 12/23/84,
12/24/84, 12/25/84, 12/26/84, 12/27/84, 12/28/84,
12/29/84, 12/30/84, 12/31/84.

AD INDEX

Anirog.....	IBC
A P S	69
Audiogenic	IFC
Calco	90
Honeyfold.....	42
Llamosoft	23
Pitman Publishing	88
Quicksilver.....	72
Richard Shepherd	79
Supersoft.....	OBC
Terminal Software	29
Trojan Products	88

ANIROG

**NEW
TURBO 64
GAMES**

P.C. FUZZ



Aggro at closing time at the local. Getaway car screeches to a halt outside a bank. Ingenious methods employed by the Mafia to literally spirit away the loot from the High Street. Never fear - P.C. Fuzz is on patrol.

COMMODORE 64 £7.95

ZARCA MISSION



**Also available on
Disk at £9.95**

An action packed 3D scrolling arcade thriller by the author of Space Pilot

COMMODORE 64 £7.95

TRADE ENQUIRIES: ANIROG SOFTWARE LTD., 39 WEST HILL, DARTFORD KENT (0322) 92513/8
 MAIL ORDER: 8 HIGH STREET, HORLEY, SURREY. 24 HOUR CREDIT CARD SALES: HORLEY (03304) 6083
 PAYMENT BY CHEQUE: P.O. ACCESS/VISA, 50p POSTAGE & PACKAGING

BUSICALC 3

- the sophisticated spreadsheet !

Easy to learn, easy to use - something that can't be said of many business programs. But it's true of all the programs in the BUSICALC series.

BUSICALC 3 can handle all sorts of jobs - budgets, expenditure analysis, stock lists, price lists, and product costing are just a few of the possibilities. Three-dimensional formulae automatically access data stored on disk, so that you can easily pull together information from several different sheets and summarise or manipulate it.

It's simple to transfer data to other programs such as EasyScript. And you can use virtually any printer with BUSICALC 3, whether dot matrix or daisy wheel, Commodore or non-Commodore.

For the CBM 64 and PET/CBM 4000 & 8000 series.

Available through dealers or from:

Supersoft, Winchester House, Canning Road, Harrow HA3 7SJ

Phone 01-861 1166 for more details and a free catalogue.

